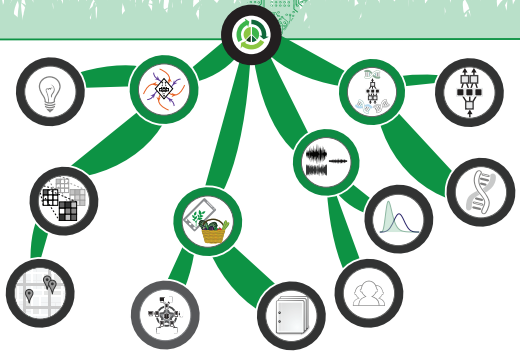


Supporting Sustainable Decision-Making with Value-Sensitive Artificial Intelligence

Doctoral Thesis Manuscript by
Thomas Asikis
Diss. No. 28053

DISS. ETH NO 28053

SUPPORTING SUSTAINABLE DECISION-MAKING WITH VALUE-SENSITIVE ARTIFICIAL INTELLIGENCE

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)
presented by

THOMAS ASIKIS
Master of Science in Information Systems,
Athens University of Economics and Business,
born on 11 August 1991,
citizen of Greece

accepted on the recommendation of
Prof. Dr. Dr. h. c. Dirk Helbing, examiner
Prof. Dr. Petros Koumoutsakos, co-examiner
Prof. Dr. Avishek Anand, co-examiner

2022

To my parents

Abstract

Sustainability is a term that is becoming increasingly prevalent, as several recent catastrophic events are often attributed to the impact that modern lifestyle has on the environment. Moreover, the term extends also to sustainability of operational democratic societies, where availability and accessibility to several critical infrastructures is ensured for individuals. Yet, achieving sustainability can be challenging. Recent reports from the United Nations conclude that improving decision-making processes on several levels, from institutional level policy making to individual level everyday decisions, supports sustainable development. Typically, deciding on more "sustainable" solutions often leads to complex multi-objective optimization problems, which are not trivial to solve. Modern artificial intelligence (AI) provides several methods to handle complex problems, particularly within the fields of machine learning and optimization. Nevertheless, AI has also given rise to new challenges, especially related to privacy, autonomy, and personal values and morals. Thus, the need for value-sensitive AI systems arises, where values and preferences are included directly in the system design.

The current thesis pursues the paradigm of efficient value-sensitive AI, mainly focusing on sustainable decision-making problems, by providing experimental, empirical, and methodological arguments. Three main design approaches are considered: centralized, decentralized, and a hybrid combination of both. First, an AI system applies centralized controls to simulations of critical infrastructure components, such as power grids. The results show the ability of said AI controls to stabilize and sustain the operation of critical infrastructures. Yet, centralized approaches may restrict and suppress personal freedoms and autonomy, especially when applied on individuals. Thus, a decentralized approach is evaluated next in the domain of sustainable product recommendations. The proposed AI system receives explicit input from individuals regarding their morals and values, and then calculates personalized ratings for sustainable products. Interactions between the decentralized system and individuals are evaluated in a field-study on two grocery stores. Statistically significant results confirm the system's ability to support individuals towards more sustainable purchases.

Finally, hybrid combinations of centralized and decentralized approaches are evaluated. A novel privacy-preserving framework is proposed to calculate accurate aggregations of individual data without exposing the actual individual data to centralized systems. Additionally, a novel hybrid AI system is introduced and combined with the privacy-preserving framework to generate sustainable basket recommendations based on personal values and environmental objectives. Quantitative results on a synthetic dataset show a considerable reduction of environmental impact, even when users adopt only a fraction of the recommendations.

Zusammenfassung

Nachhaltigkeit ist ein Begriff, der zunehmend an Bedeutung gewinnt. Dies aufgrund immer häufiger vorkommenden, katastrophalen Umweltereignissen, welche auf die Auswirkungen des modernen Lebensstils zurückzuführen sind. Darüber hinaus erstreckt sich der Begriff auch auf die Nachhaltigkeit funktionierender demokratischer Gesellschaften, in denen die Verfügbarkeit und Zugänglichkeit verschiedener kritischer Infrastrukturen für jeden Einzelnen gewährleistet ist. Jedoch ist das Erreichen von Nachhaltigkeit eine Herausforderung. Aktuellste Berichte der Vereinten Nationen kommen zum Schluss, dass eine Verbesserung der Entscheidungsprozesse auf verschiedenen Ebenen, von der institutionellen Ebene der Politik bis hin zu den alltäglichen Entscheidungen eines jeden Einzelnen, eine nachhaltige Entwicklung unterstützen könnte. Typischerweise führen "nachhaltigere" Lösungen jedoch oft zu komplexen, multi-dimensionalen Optimierungsproblemen, die nicht mit einfachen Mitteln zu lösen sind. Moderne künstliche Intelligenz (KI) bietet verschiedene Methoden zur Lösung komplexer Probleme, insbesondere in den Bereichen maschinelles Lernen und Optimierung. Dennoch hat KI auch einige Herausforderungen mit sich gebracht, insbesondere in Bezug auf den Schutz der Privatsphäre, der Gewährleistung von Autonomie und der Achtung persönlicher Werte und Moralvorstellungen. Deshalb steigt der Bedarf nach wertsensitiven KI-Systemen, bei denen Werte und Präferenzen direkt in das Systemdesign miteinbezogen werden.

Die vorliegende Dissertation verfolgt dieses Paradigma einer effizienten und wertsensitiven KI. Mit Fokus auf Entscheidungsfindungsproblemen im Bereich der Nachhaltigkeit, werden experimentelle, empirische und methodologische Argumente, Resultate und Lösungen aufgezeigt. Es werden drei Designansätze betrachtet: Zentralisiert, dezentralisiert und eine hybride Kombination aus beiden Ansätzen. Zunächst wendet ein zentralisiertes KI-System Kontrollen in Simulationen kritischer Infrastrukturkomponenten an, wie z.B. Stromnetzen. Die Ergebnisse zeigen, dass diese KI-Kontrollen in der Lage sind, den Betrieb kritischer Infrastrukturen zu stabilisieren und aufrechtzuerhalten. Zentralisierte Ansätze können jedoch persönliche Freiheiten und Autonomie einschränken und gar unterdrücken, insbesondere wenn sie auf Einzelpersonen angewendet werden. Daher wird als nächstes ein dezentralisierter Ansatz im Bereich der nachhaltigen Produktempfehlungen evaluiert. Das vorgeschlagene dezentralisierte KI-System erhält von Einzelpersonen Informationen zu ihren Moralvorstellungen und Werten und berechnet darauf basierend personalisierte Empfehlungen nachhaltiger Produkte. Die Interaktionen zwischen dem dezentralisierten System und Einzelpersonen werden in einer Feldstudie in zwei Lebensmittelgeschäften untersucht. Statistisch signifikante Ergebnisse bestätigen die Fähigkeit des dezentralisierten KI-Systems Individuen zu nachhaltigeren Einkäufen zu bewegen.

Schliesslich werden hybride Kombinationen von zentralisierten und dezentralisierten Ansätzen evaluiert. Ein neuartiges Framework zur Wahrung der Privatsphäre wird vorgeschlagen, um genaue Aggregationen individueller Daten zu berechnen ohne die tatsächlichen individualbasierten Daten an zentralisierte Systeme weiterzugeben. Darüber hinaus wird ein neuartiges hybrides KI-System eingeführt und mit dem datenschutzfreundlichen Framework kombiniert, um nachhaltige Warenkorbempfehlungen auf der Grundlage persönlicher Werte und ökologischer Ziele zu generieren. Quantitative Ergebnisse aus einem synthetischen Datensatz zeigen eine beträchtliche Verringerung der Umweltbelastung, selbst wenn die User nur einen Bruchteil der Empfehlungen annehmen.

Acknowledgements

The completion of a Doctoral thesis is a monumental yet colossal task. The current thesis is the outcome of countless hours of collaborative work, discussions, and creative interactions not only during the doctoral program, but also before. I consider the thesis to be the outcome of acquisition of proper skills, experience, and knowledge throughout all my life. For this reason, I would like to thank all the people that helped me acquire all the necessary tools and supported me through my everyday life.

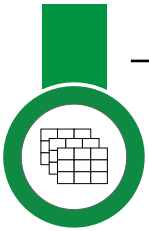
First, I would like to thank my family, who always supported me in the journey of seeking knowledge not only by providing me all the necessary resources for a comfortable living, but by also introducing me to the fundamental concepts and values of science and research from a very young age. Without my parents and sister I would have faced many more challenges and struggles in order dedicate myself to research. Furthermore, I would like to thank my niece and nephew, who brought happiness in our family and inspired me to pursue a more sustainable future for their generation. Finally, I would like to thank all the people that were there next to me as a friend and a partner, mainly: Jennifer, Alex, Thanos, Panos, George, Andreas, Giannis, Sotiris, et al.

Next, I would like to thank Prof. Dirk Helbing as my doctoral supervisor for providing me with continuous guidance and input, while trusting in my abilities. Furthermore, I would like to thank my academic collaborators, for their invaluable input and support. I would like to thank Dr. Nino Antulov-Fantulin and Prof. Lucas Böttcher for supporting me extensively during the last years of the doctorate and allowing me to further improve my knowledge and understanding of complex networks and physics. I am very grateful to Evangelos Pournaras, who supported me tremendously during the doctorate studies and provided me with informative input in value-sensitive design of information systems for sustainability. I would also like to thank the ASSET consortium and Johannes Klinglmayr, where I had the chance to see my models and implementations make a difference in the real-world.

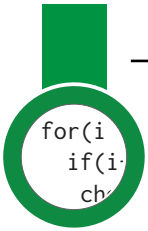
Several insightful discussions with my academic peers helped me extend my knowledge and horizons. I would like to thank my COSS fellows, the two Stefanos, Farzam, Caleb, Stefan, Mark, and Jovan for the good times and creative academic experiences during the COSS retreats. I need to thank especially Heinrich, Jennifer, the DeScil lab, and the tutors and teaching assistants of ETH that helped me understand and apply my work, especially on causal inference, game theory, statistics, and deep learning. I want to say thank you to all the office managers who supported me during my stay at COSS. I also want to thank all the people provided me with guidance and support during my professional and academic career as co-workers, co-students, and teachers, especially professors G. Lekakos, I. Androutsopoulos, C. Tarantilis, and Dr. K. Tsagkaris.

Contents

List of Figures



List of Tables



List of Algorithms

Chapter

1

Introduction

Every day we are urged to take decisions to reach our personal or collective goals. Often, a person, a community, or even an entire society evaluates past decisions in retrospect to examine counterfactual scenarios. Would a different outcome have helped to reach a goal with lower costs or less undesirable side effects? Would another goal allow for better decision-making and the ability to pursue more important goals today? Do the goals and our decisions lead to a brighter future or a catastrophe? Such questions are often difficult to answer, even in hindsight. However, the attempt to answer challenging questions drives human progress in science and technology.

Personal goals and decisions can conflict with the goals of other individuals and the collective [1, 2]. When such conflicts appear, individuals often prioritize their personal goals. Then, the so-called *tragedy of the commons* [1] may arise. Often, in tragedies of the commons, the long-term negative consequences of failing to reach collective goals outweigh the initial satisfaction of achieving personal goals. Examples of such tragedies include the over-consumption of water in arid lands, the unwise harvesting of trees to meet the demand of lumber and paper markets, noise and pollution caused by the usage of cars, the adoption of unhealthy diets, or the purchase of non-fair trade products due to lower prices [3]. Eventually, ignoring collective goals leads to extensive consumption and pollution. Environmental catastrophes, social inequalities, and adoption of unhealthy lifestyles are common outcomes of such tragedies [3, 2, 1]. For example, individuals consume high amounts of energy in their everyday lives to achieve more comfortable lifestyles. Energy consumption increases the usage of fossil fuels for its production [4]. The usage of fossil fuels is the primary cause of climate change that can lead to observed extreme weather phenomena, such as cyclones [5] and heatwaves [6]. As extreme weather phenomena become more common, environmental catastrophes also occur more frequently. In fact, we already observe an increasing frequency of catastrophic events, such as wildfires [7, 8], floods [9, 10], and desertification [11, 12]. Climate change and severe weather phenomena can create considerable social challenges, such as large population displacements [13]. The continuous common occurrence of such phenomena may endanger sustainable existence of operating democratic human societies. If individuals were provided and communicated better alternative ways to manage their energy consumption, perhaps such catastrophes could be avoided.

Therefore, the ability to understand future effects of our decisions and goals is paramount for the long-term sustainable development of human societies.

1.1 Sustainable Decision-Making

Sustainability is considered the ability to meet current individual and collective goals and needs without compromising collective future goals and needs [14], e.g. by avoiding the aforementioned catastrophes. The outcomes of collective and individuals' behavior may accumulate and lead to future *risks* and *existential challenges*, such as irreversible environmental damage, health crises [15], loss of freedom and autonomy, and social unrest [16, 17]. A main driving factor of several existential challenges are the decisions that lead to unsustainable resource consumption and allocation. From a policy making and organization perspective, miss-management of critical infrastructures may lead to highly costly black-outs and severe socio-techno-economical challenges [18]. From an individual perspective, consumer needs and preferences are satisfied by the production and distribution of a great variety of market products, e.g. food products [11], which require vast quantities of energy and raw materials. Food supply chains have a considerable ecological footprint and can lead to future shortages of water, raw materials, desertification, and even forced mass migration [19, 11]. Addressing challenges arising from *resource consumption* is important to achieving sustainable development while respecting individual needs.

Many institutes and researchers [20, 21, 22] study the mechanisms between present and future goals and needs. One of these studies focuses on the United Nations (UN) Sustainable Development Goals (SDGs) [20]. The UN SDGs reports propose, analyze, and identify everyday tasks that can support sustainable development by assisting organizational and individual decision-making. According to the UN, the SDGs comprise 134 targets [23] for sustainable development. Figure 1.1 illustrates an overview of the SDGs. Nevertheless, synergies and trade-offs between SDGs and personal needs cause challenging dilemmas regarding the goal prioritization [25], which make sustainable decision-making challenges non-trivial. Existing effort towards sustainable development appears to be contradicting with the way people decide to consume resources [11]. In particular, decisions on sustainable resource allocation that satisfy both present and future individual and collective goals often lead to complex decision problems [26, 27, 28]. For example, deciding upon the optimal carbon dioxide tax and constraints on human interventions in ecosystems. Successful sustainable decisions are often the result of solving *high-dimensional optimization problems* concerning dynamical systems, which typically include models of weather, economy, and society [29]. Such optimization problems have constraints and objective functions related to complex system dynamics and personal and collective needs [26, 30]. For example, a common optimization approach in (socio-)economics is the maximization of utility and welfare functions *over a long period of time* [31].

In this thesis, decision-making is considered as the identification of an optimal set of actions or courses of actions in order to tackle complex multi-dimensional sustainability problems [32]. Individual decision-making and organizational policy-

Figure 1.1: The Sustainable Development Goals proposed by the UN [24].

making can be supported by identifying and suggesting such optimal actions. It is important to point out the relevance of AI optimization methods to decision-making both in the fields of management science [33] and social sciences [34], where experts often use such methods to consider a rational course of actions. Choosing among optimal decisions derived from analytical or heuristic models can also be considered as rational [35, 34, 36] or analytical decision-making [33]. Non-rational and intuitive decision-making, where the individuals often choose non-machine proposed solutions, are also compatible with the proposed frameworks. In the current thesis, the decision-maker can compare AI generated solutions with intuitive solutions and then decide the appropriate course of actions.

1.2 The Role of Modern Artificial Intelligence

Artificial intelligence (AI) can tackle the difficult *optimization problems* related to sustainable decision-making [23] by offering many useful tools, such as predicting sustainability indicators with NNs [37] (NNs) or optimizing actions that limit environmental impact with genetic algorithms [38]. From a technical perspective, decision optimization and predictive analytics support decision-making across various fields, such as economics, game theory, and engineering, especially for decision problems that involve many input variables [32]. Optimal control and reinforcement

learning are also often considered as potential modelling tools for deciding subsequent actions that involve long term outcomes [39]. Applications of AI methods can assist sustainable decision-making both at individual [40] and collective[1] [37, 38] level. Nevertheless, methods that combine individual and collective requirements to achieve sustainable decisions in all levels, are sparse in literature. From AI related optimization methods, optimal control has been considered as a solution to sustainable decision-making, e.g. theoretical analysis of potential strategies for minimization of carbon dioxide emissions [41] and extensions in Integrated Assessment Models (IAM) [42]. International organizations and governments use optimal control outputs in simulations for policy making regarding sustainable development, such as the "Dynamic Integrated Climate Economic" (DICE) model [29]. Applications of optimal control for sustainable societies are also common in socio-economics, especially in the research field of utility-welfare systems [31].

Another key application of AI is *representation learning* [43] and *knowledge organization* [44]. These methods can organize and provide knowledge to create informative representation and ontologies [44], which can process, combine, summarize, and represent a high volume of unstructured data from various disciplines [23]. Modern knowledge databases combine machine learning methods in tandem with human expert knowledge. Knowledge bases include automated procedures to transform data into easily accessible and understandable information [45]. Automating parts of the data processing and knowledge generation with machine learning can lower operational and maintenance costs that occur in pure white-box expert or knowledge base systems [46]. Machine learning systems provide high performance models that are easier to maintain due to lower human capital costs in data acquisitions and processing [47]. Recent advances in natural language [48] and image processing [49] further enhance knowledge bases by processing complex unstructured data from images and text.

Smart and explainable representations can be used both for personalization, but also for clarifying individual doubts regarding SDGs. Diverse teams of experts collaborate in order to process this data into useful knowledge that is later used by expert systems for recommendations [50]. Success in achieving SDGs requires collaboration among individuals, who are mainly non-experts. Therefore, it is essential to provide knowledge around SDGs in an accessible and understandable manner to individuals [11]. Once public access is possible, we need to evaluate and understand the effect of such knowledge on individuals [51]. For example, individuals might be more willing to adjust their personal goals and accept stricter constraints in their consumption once they realize the effects of failing to achieve sustainability goals [52]. Thus, organizing and processing data into accessible knowledge [53] is a paramount task to understand the dynamics between individuals and achieving sustainability goals.

Finally, recommender systems and learning assistants use *knowledge representation* to calculate *optimal personalized sustainable recommendations* for individuals [54, 55, 56]. Recommendations can support individuals in complex decision problems and also improve their understanding of the effects of such decisions on achieving

[1] e.g. organizational or institutional decisions.

sustainability goals [57]. Policy making and sustainable societal implementations can also benefit from recommendations, which may provide experts with important knowledge regarding policy feasibility, acceptance, and potential outcomes [58]. Concluding, the design and implementation of recommender systems can benefit us in making more sustainable decisions.

1.3 Value-Sensitive Design to Improve Artificial Intelligence

AI provides high performance tools to support sustainable decision-making, such as recommender systems. But most of these systems often focus on customer retention, market share increase, and profit maximization [59, 60, 61]. Furthermore, those systems are costly, centralized, and non-transparent, meaning that they are inaccessible to the majority of the public causing inequalities amongst users [51]. Another potential caveat is that AI systems are usually closed and proprietary. Proprietors limit access and user intervention to the system components, thus, true personalization may not be possible. Many machine learning applications rely on supervised learning techniques that require large amounts of data to generalize efficiently [62]. An individual might not have access to the required amount of data, and techniques like data augmentation [63, 64] may not always help. Even when individual data is enough for training AI models, challenges regarding privacy and security arise [16, 17, 65, 66, 51].

SDGs propose several frameworks for lifestyle changes [11], yet there is still little adaption of more sustainable lifestyles by individuals [67]. *Communication* and *clarification* can help to make SDGs more understandable to individuals. Furthermore, combining existing personal goals and SDGs to determine new personal goals may substantially support adaption of sustainable lifestyles. Recommending non-invasive changes that are compatible with each individual's unique lifestyle is essential for achieving optimal *personalized* sustainable decisions.

Value-sensitive design combines individual values, e.g. privacy or taste preference, along with collective values arising from the SDGs in order to design and implement AI systems that support sustainable decision-making [68, 69]. The key concept of value-sensitive design is the extraction of the individual stand on the aforementioned values. Intuitive user interfaces, knowledge extraction, and representation technologies can extract and store data relevant to determine sustainability problem constraints and objectives from user-provided data [19]. Self-determination is given when the users explicitly provide information on their personal values and receive feedback about how following them affects individual and collective sustainability goals [19, 70]. While this process is repeated through time, a learning effect develops where individuals are free to explore and determine how their values, goals, and decisions can improve and contribute to achieving collective sustainability.

Recommender systems can support individuals in adopting a more sustainable lifestyle via their purchase preferences [22, 19]. Nevertheless, modern personalization methods, e.g. recommender systems, often rely on big data methods. Individuals

might not be able to influence the suggestions of such recommender systems explicitly, thus becoming prone to *nudging*, which causes loss of user autonomy [71]. Hence the usage of current big data Information and Communications Technology (ICT) systems, especially of systems that provide information and recommendations regarding purchases, may promote behavioral control instead of supporting individuals in achieving sustainability goals [72, 16]. Thus, the need for self-determined personalization arises, such that an individual becomes empowered to understand and choose the means and information sources to guide their own consumer behavior in everyday decisions. Complementary, individuals need to be aware of the sustainability goals in order to use effectively the proposed AI and make decisions that adhere to the SDGs.

1.4 Positioning in Literature

The main focus of this thesis is to develop and enhance AI systems in supporting decisions that achieve both personal and collective goals related to environmental and societal sustainability by introducing a value-sensitive design layer [73]. As Figure 1.2 illustrates, this thesis lies in the intersection of three research fields, namely: AI, value-sensitive design and sustainable decision-making. In contrast to most existing related work, the current thesis integrates theory, experimental design, simulation, and data analysis to identify and utilize features and criteria that can support sustainable decision-making. Recent research articles [23, 11, 51] focus mostly on theoretical aspects and guidelines on potential applications of responsible AI to sustainability related goals. The current thesis follows the aforementioned guidelines and implements such system design, while evaluating the proposed AI systems in terms of individual acceptance, model performance, and efficiency. From a theoretical perspective [23, 74], AI can tackle intractable analytical problems and challenges arising when pursuing the SDGs. Recapping the related work [23], the key driving factors causing AI to become an inhibitor revolve around (i) having unequal access to AI,(ii) privacy intrusion, which can lead to loss of freedom and autonomy, and (iii) energy consumption by the use of computing infrastructure. The current thesis offers a comprehensive experimental design that evaluates the outcomes of designing and using value-sensitive AI solutions based on the three aforementioned factors. Such outcomes align the current thesis towards value-sensitive design [73] and namely responsible AI [51], which is the intersection between value-sensitive design and AI in Figure 1.2. Furthermore, this thesis provides experi-

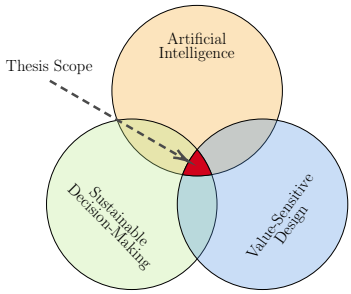


Figure 1.2: The current thesis focuses in the intersection of 3 major research fields, namely AI, value-sensitive design, and sustainable decision-making.

mental and applied analysis to fill in the gap between theory and application. More specifically, contributions of this thesis evaluate three different AI design approaches, where a system consists of individual elements and actors that may require AI input to make more sustainable decisions:

- Centralized, where the AI decision support system is a single entity, which has access to all the system data/observations and supports decision-making of elements and actors that require AI input.
- Decentralized, where an independent decision-making support AI is paired with each element or actor that requires AI input. The paired AIs support decision-making of each element, without communicating with a centralized system.
- Hybrid, where centralized and decentralized AI decision-making support systems co-exist and communicate with each other to determine proposed decisions.

The centralized and decentralized decision-making terms extend and combine definitions used in existing literature from both organizational theory [75] and algorithmic decision-making [76]. Centralized design approaches are more preferable on an institutional level, e.g. sustainable policy making and emission regulations, where decisions are applied in a large scale and are often long term. Decentralized approaches are preferable on an individual level, where individual freedom, preferences, and acceptance need to be respected in order to achieve the goals, such as making food consumption more sustainable. Ultimately, hybrid approaches can be developed by extending and combining the decentralized and centralized systems developed in this thesis. Hybrid approaches allow for integration of centrally generated information, without necessarily limiting individual freedoms in the decision process.

The combination of AI and value-sensitive design also requires substantial theoretical, experimental, and applied research regarding data-driven design. The analytical modeling of techno-socio-economic systems often uses and combines various experimental and theoretical approaches [70, 77, 26, 30]. Typical examples of such modeling include the derivation of optimal controls for economic and supply demand systems [78, 41] and technical infrastructure [77], which often introduce analytical and computational simplifications in order to reduce complexity and derive tractable analytical solutions. The current thesis aims to reduce potential simplification issues by tackling less simplified versions of complex sustainability problems. Control and optimization algorithms are proposed in this thesis that achieve state-of-the-art performance in both existing and novel control and (multi-objective) optimization tasks. The performance, analysis, and evaluation of those algorithms actively contributes to control, machine learning, and optimization literature.

In particular, several contributions of the current thesis are relevant to recommender system literature. A recommender system complexity may further increase as value-sensitive design integrates individual constraints and objectives to sustainable decision problems. This challenge is one of the key considerations during the development of the recommender algorithms presented in this thesis. Users exercising self-determination want recommender systems to be fast and responsive [79] to test recommended and personal decisions under different combinations of personal goals

and constraints. When an AI system requires large amounts of time to provide recommended decisions, an individual will try fewer combinations and ultimately receive less information to make an educated decision. Furthermore, it is essential to calculate highly optimal recommendations both in terms of collective and personal criteria. Individuals may reject poorly personalized recommendations [80] or adopt personalized recommendations with high environmental costs. If individuals adopt poorly personalized recommendations that cause dissatisfaction, individual trust in the AI system [81, 82] and system effectiveness in the long-term will also decrease. Moreover, modern recommender systems operate mostly independently, where individual choices are independent of the choices of others [40, 50] in terms of sustainability. In contrast, resulting recommender systems of the current thesis pursue collective goals that can be decided both collectively by aggregating individual goals in a privacy-preserving manner and by consulting theory and experts.

In the research articles corresponding to the chapters of this thesis my co-authors and I: (i) identify individual and collective goals regarding the SDGs, specifically with sustainable consumption (Chapter 3), (ii) design and implement AI frameworks that can support complex decision-making tasks such as control of critical infrastructures or improving sustainable consumption (Chapters 2, 4 and 6), (iii) showcase and quantitatively evaluate models and potential applications that improve decision-making and sustainable development (Chapters 3 and 4), and finally (iv) design and implement privacy-preserving aggregation mechanisms that allow for privacy-preserving and value-sensitive calculation of metrics related to collective goals from individual values (Chapter 5).

1.5 Thesis Overview

In this *cumulative thesis*, all three potential AI designs are evaluated to determine their effectiveness on relevant tasks. From application perspective, two main types of AI systems are considered, namely systems that: (i) calculate controls, which preserve and maintain critical infrastructures in a centralized manner, such as power grids (mainly SDG 7 "Affordable and Clean Energy"[2]) and hospitals (mainly SDG 3 "Good Health and Well Being" 3) during crises, and (ii) enable and empower individuals to make more sustainable decisions during their everyday lives by adopting more sustainable consumption patterns (mainly SDG 12 "Responsible Production and Consumption"). The main working pipeline in this thesis collects data from simulations, individuals, and organizations, combines the data to form a knowledge base and/or implements a relevant computational model. Novel AI techniques, mainly optimization and machine learning, are applied on such models to provide individuals and system shareholders with information that supports sustainable decisions. This thesis comprises two parts. Part I focuses on system design and applications on sustainable decision-making. Part II provides a more technical perspective on the techniques developed for privacy optimization and NN control used in the first part.

[2] see Figure 1.1

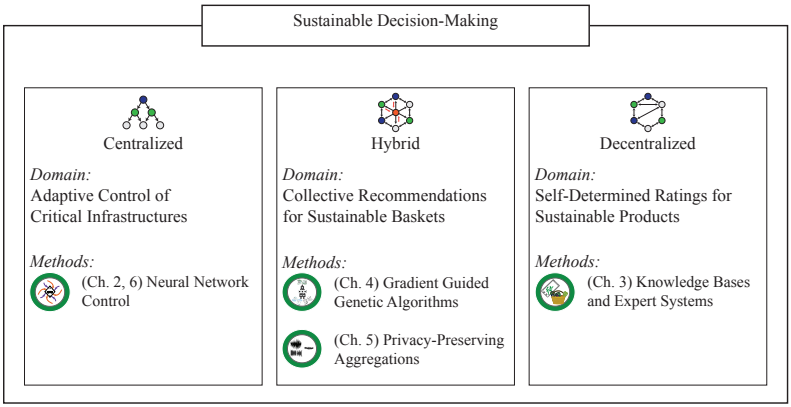


FIGURE 1.3: High level overview of the thesis chapters. The overarching focus is sustainable decision-making and three main design approaches are evaluated on representative sustainable decision-making problems. To attempt a more systematic approach to the design of value-sensitive AI for support of sustainable decision-making, all AI design paradigms (centralized, decentralized, and hybrid) are considered.

Figure 1.3 provides a high level overview of the design approaches considered in this thesis. Chapter 2 evaluates a centralized AI algorithm, which calculates systemic interventions (control signals) by training and evaluating NNs on analytical simulations of systems that can model critical infrastructures. The centralized control methods are further evaluated from a technical perspective in Chapter 6. Next, a decentralized approach is evaluated in Chapter 3, where the proposed AI system is designed and implemented to support sustainable decisions in daily food consumption with respect to consumer needs and values. The proposed system relies on expert systems and knowledge bases and is tested in a field-study in two European retailer stores. Both of the above systems lack a privacy-preserving mechanism that allows implementation and mixing of both decentralized and centralized approaches. Therefore, Chapter 5 proposes a hybrid optimization framework to collect and aggregate individual data in a self-determined privacy-preserving manner. Accurate data aggregations can be used by organizations and stakeholders of systems similar to Chapter 2 to provide insights when institutions decide policies and disseminate information to individuals. Such information can be used by decentralized systems to further enhance the support of sustainable decision-making. Finally, Chapter 4 proposes a multi-objective framework based on a combination of the computational methods discussed in Chapters 2 and 3. Initially, the proposed framework is designed in a decentralized manner and a hybrid design is developed by taking advantage of the privacy-preserving mechanisms of Chapter 5. Representative domain applications are evaluated and novel AI techniques are proposed by extending and combining existing AI methods.

Chapter 2 introduces a novel NN control method, which is called "Neural Ordinary Differential Equation Control" (NODEC). NODEC is evaluated on control tasks on higher-dimensional non-linear ordinary differential equations (ODEs). The evaluation of NODEC is performed on Kuramoto oscillators, which are often used for modeling controls in power grids [83, 77] and disease spreading dynamics [84] to determine effective countermeasures to stop the spread by resource allocation. Non-linear ODEs can model several real-world systems, such as disease dynamics, power grids, biological systems, economic phenomena [85, 86, 87, 88]. A comparison of NODEC with popular control baselines confirms its ability to tackle difficult control problems, while requiring low control energy. Control energy is relevant to the total absolute value of the control signal applied. High energy controls imply more invasive and high-impact interventions in the controlled system. Furthermore, NODEC is shown to work under limited access to system components, e.g. controllability of nodes can be decided in a participatory manner, and thus, not all nodes are controllable. Forced controls that apply to each and every individual element are unwanted in dynamical systems that represent socio-technical or ecological systems [26], as they might be considered authoritative and oppressing for individuals represented by the nodes. The contents of Chapter 2 are accepted for publication as an article titled "Neural Ordinary Differential Equation Control of Dynamics on Graphs" in the American Physical Society Journal *Physical Review Research*.

Chapter 3 addresses sustainability related goals in a decentralized approach for phenomena that affect individual freedom and autonomy,. Applications of centralized control frameworks may not always be compatible with *social norms, individual preferences, and autonomy* [51]. Ideally, AI should contribute to the development of free and democratic societies, and not to the rise of technocratic autocracies. Therefore, alternative approaches may need to be developed that take into account individual and collective constraints. Thus, centralized controls may not be acceptable by individuals, when the problem objectives and constraints involve personal freedom. In contrast to Chapter 2, which focuses on analytical models and centralized AI, Chapter 3 mainly focuses on decentralized AI systems. Furthermore, in Chapter 3, experts and crowd-sourcing contribute to the construction of a knowledge base of collective and individual goals and corresponding product characteristics that help to achieve them. Next, a recommender system is developed, which operates on the knowledge base calculating sustainability ratings for products. The proposed recommender system combines individual input with expert rules to provide recommendations for sustainable purchases in the form of smart ratings for each product. Smart ratings consider four types of sustainability criteria: individual health, environmental impact, social impact, and product quality. A field study in two European retailers is used to provide statistical evidence of the *effectiveness* of the proposed recommender system in a *real-world* scenario. Causal impact analysis [89] on the outcomes of the study shows that SDG-oriented self-determined recommender systems increase expenditures towards more sustainable products. The chapter includes important insights for the design and implementation of AI systems developed for a similar application in Chapter 4. The main outcomes of Chapter 3 were published [90] in the journal of *Royal Society Open Access* in an article titled:

"How value-sensitive Design Can Empower Sustainable Consumption". This paper received the best paper award from the Global Food and Environment Institute of the University of Leeds[3].

Chapter 4 introduces a decision-making problem, in which a consumer is provided with a set of recommended baskets that satisfy their personal preferences and also minimize environmental impact. The design of the optimization problem in Chapter 4 expands the recommender constraints and objectives discussed in Chapter 3 by conducting a more quantitative study on a synthetic dataset, which uses existing open data [91, 92, 11]. Existing baselines and a novel evolutionary algorithm, termed gradient guided genetic algorithm (G3A), are proposed to tackle the problem initially in a decentralized manner. The algorithms and design paradigms introduced in Chapters 2 and 3 are combined in this chapter. A hybrid design of G3A is then proposed by introducing collective constraints in the recommendation process. The collective constraints are calculated in a centralized privacy-preserving manner as discussed in Chapter 5. Simulation results using the synthetic dataset showcase the ability of multi-objective optimization algorithms to provide personalized recommendations that achieve considerable reduction of environmental impact of purchases. A comparison of the hybrid and decentralized implementations of G3A indicates that incorporating collective information in a hybrid approach can potentially further reduce the environmental impact of the recommended baskets. Chapter 4 is a sole authorship work and is currently being finalized for submission as an article titled "Multi-Objective Optimization for Value-Sensitive and Sustainable Basket Recommendations".

Chapter 5 proposes a novel method for privacy-preserving aggregations, which can determine aggregate values of individual data. Applications such as the smart ratings of Chapter 3 allow for an effective decentralized design, but still lack the privacy-preserving mechanisms to collect accurate data aggregations that can be used for macroscopic studies of the systems and complementary centralized implementations. Ideological preferences regarding sustainability and consumption data are personal and their use and processing may be unwanted from individuals due to privacy concerns. Collection of such data is also challenging under new legislative frameworks such as the General Data Protection Regulation (GDPR) [93]. Chapter 5 offers a new solution to the problem backed by both analytical arguments and experimental evaluation. Sensor or individual data are masked by using a set of noise generating mechanisms that follow the differential privacy paradigm [94]. The masked data are difficult to be mapped to their original values, while the aggregate of the original values is effectively approximated by aggregating the masked values. For illustration, experiments are performed on consumer data of a power consumption dataset [95]. The addition of differentially private noise introduces new challenges for the accurate calculation of collective goals. Hence, the trade-off between privacy and accuracy is studied extensively in this chapter. The contents of this chapter are published in the Elsevier journal [96] of *Future Generations Computer Systems* under the title: "Optimization of Privacy-Utility Trade-offs under Informational Self-determination".

[3] `https://www.leeds.ac.uk/global-food-environment-institute/doc/awards`, accessed October 2021.

Chapter 6 complements Chapter 2 by evaluating time-dependent controls with a NODEC framework, termed AI Pontryagin. Chapter 6 contains analytical arguments and experimental evaluation show that AI Pontryagin has implicit energy regularization properties, achieving similar performance to optimal control baselines, which theoretically achieve the lowest energy controls to steer the system to a desired target state. AI Pontryagin is tested in both linear and non-linear systems and on different graph-structures, further confirming that NODEC can achieve low energy controls and near-optimal behavior. The contents of Chapter 6 are accepted for publication as an article titled "AI Pontryagin or: How Artificial Neural Networks Learn to Control Dynamical Systems" in the Springer Journal *Nature Communications*.

All chapters of the main thesis have a corresponding Appendix, sharing the same title. Appendix Chapters are indexed with a capital latin letter, whereas main thesis chapters are indexed with numbers. In each Appendix there is a nomenclature table, which supports the reader by providing descriptions for most symbols appearing in the corresponding thesis chapter. The Appendices include technical details, such as parameters, training procedures etc. Data and code availability statements are provided in the appendices. Code repositories may contain more extensive technical descriptions and information about parameterization. Finally, a table containing most abbreviations for all chapters can be found in the appendix Chapter F.

Part I

Design of Value-Sensitive AI for Sustainability

Chapter

2

Centralized Control on Critical Infrastructures with Neural Ordinary Differential Equation Control

From AI related optimization methods, optimal control has been considered as a potential centralized solution to sustainable decision-making, e.g. theoretical analysis of potential strategies for minimization of carbon dioxide emissions [41] and potential extensions in Integrated Assessment Models (IAM) [42]. International organizations and governments use optimal control outputs in simulations for policy making regarding sustainable development, such as the "Dynamic Integrated Climate Economic" (DICE) model [29]. Applications of optimal control for sustainable societies are also common in socio-economics, especially in the research field of utility-welfare systems [31]. The field of reinforcement learning (RL) also offers applications for sustainability-related decision problems [97]. Environmental and socio-economic optimization problems often involve several high-dimensional non-linear dynamics with non-convex optimization objectives and constraints [27, 28].

Typically, control of critical infrastructures to prevent cascading failures is often studied within the field of optimal control of power grids [83, 87, 77]. Other critical infrastructures, such as health and supply systems can be modelled by complex continuous time dynamics [84, 98]. Calculating optimal controls that provide subsequent decisions can enhance the performance of such infrastructures and also avoid potential failures of such systems. Preserving accessible and operational infrastructures is relevant to many sustainability goals [24], and, thus, it is important to consider optimal solutions that reduce the impact of infrastructure failures during decision-making. In particular, applications of gradient based control frameworks, such as the one discussed in this chapter, can be applied to a multitude of SDG related problems that are described by ODE systems. For example, ensuring healthy lives and promoting well-being for all at all ages by effective budget and expenditure control in disease spreading [84] is aligned with SDG 3 "Good Health and Well Being" [24]. Ensuring access to critical infrastructure by control is also highly relevant to several UN SDGs that evaluate operation and accessibility of such infrastructures, namely SDG 7 "Affordable and Clean Energy" which aims to optimize access to affordable energy and improvement of power grid efficiency, and, thus, prevention of blackouts [77]. Another potential benefit from application of control on irrigation and water supply networks [99] could be relevant to SDG 6 "Clean Water and Sanitation" that aims to ensure availability and sustainable management of

water and sanitation for everyone [24]. Building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation [24] is the main target of SDG 9 "Industry, Innovation and Infrastructure", which can benefit from control methods. Finally, optimal control [29] has been applied by Nordhaus (1993) in fundamental research on the fields of managing CO_2 emissions that supports SDG 13 "Climate Action". Thus, the importance of optimization and control methods for rational and analytical sustainable decision-making is important to consider for sustainable infrastructures and organizational policy making [35, 34, 36, 33].

Non-convex optimization problems constrained by high-dimensional non-linear dynamics can introduce serious challenges in the development of optimal control techniques [31, 100]. When solving sustainability problems, computational and analytical challenges can arise, e.g. high computation times and intractable problems. To address those challenges, complex sustainability dynamics are often simplified with techniques, e.g. linearization [101] and discrete time approximations [102, 103]. Computational and analytical simplifications may introduce considerable errors. Current optimization solutions [104, 100, 105] are under constant improvement to reduce errors. A typical example is the control of chaotic systems with recurrent NNs [106] (RNNs). Nevertheless, existing baselines often solve problems with either small time horizons, few dimensions, convex objective functions, linear constraints, and few non-linear interactions. Recent advances in the fields of "differentiable physics" and "physics informed" deep learning [107, 108] have allowed to scale deep learning prediction and control models to high dimensional systems with longer time horizons.

Dynamical processes on complex networks are common tools to model a wide range of real-world phenomena including opinion dynamics [109, 110], epidemic spreading [111, 112, 113], synchronization [114, 115], and financial distress propagation [116]. Continuous-time dynamics on complex networks can be described by different frameworks including Chapman–Kolmogorov [117], Fokker–Planck [118], stochastic differential [119], and ordinary differential [120, 121, 122] equations. The structure of many real-world systems is described by networks with certain common properties including small-world effects [123], heavy-tail degree distributions [124, 125], community structure [126], and other features [127, 128]. The control of dynamical processes on networks [129, 130] is a challenging task with applications in engineering, biology, and the social sciences [131, 132]. Optimal control signals can be calculated by solving boundary-value Pontryagin's maximum principle (PMP) problems [133, 134, 135], or computing solutions of the Hamilton–Jacobi–Bellman equation (HJB). Complementing the above approaches, we develop a Neural-ODE control (NODEC) framework that controls fully observable graph dynamical systems using neural ODEs [136]. Within this framework, FC signals are calculated by minimizing a loss function describing differences between the current and target states. We perform extensive numerical experiments on coupled high-dimensional and non-linear dynamical systems to showcase the ability of NODEC to calculate effective control signals.

Mathematically, systems are "controllable" if they can be steered from any initial state $\boldsymbol{x}(t_0)$ to any desired state $\boldsymbol{x}^*$ in finite time T. For linear systems, an analytical

condition for controllability of linear time-invariant (LTI) systems was derived by Kalman in the 1960s [137] and is known today as Kalman's rank criterion. In 1969, Popov, Belevitch, and Hautus [138] introduced another controllability test for LTI systems that relies on solutions of an eigenvalue problem of the state matrix. In the 1970s, Lin introduced the framework of structural controllability [139] as a generalization of prior definitions of controllability on graphs. More recently, different large-scale social, technical, and biological networks were analyzed from a network controllability perspective [129, 140] building on the framework introduced by Lin [139]. Controlling a complex system becomes more challenging as the number of nodes that can receive a control signal (driver nodes) decreases. Furthermore, Reference (Ref.) [141] addresses the important issue of quantifying the (control) energy that is needed to control LTI systems. Steering the dynamical system to the target state becomes even harder when energy minimization is also accounted for.

To solve general non-linear optimal control problems with energy and driver node constraints, two main approaches are used: (i) Pontryagin's maximum principle (PMP) [133, 134, 135] and (ii) Bellman's (approximate) dynamic programming [142, 143, 144, 145, 146]. Pontryagin's maximum principle [133, 134, 135] is based on variational calculus. When applying PMP, the original infinite-dimensional control problem is transformed to a boundary-value problem in a Hamiltonian framework. The downside of this approach is that the resulting boundary-value problems are often very difficult to solve. An alternative to variational methods is provided by Bellman's dynamic programming, which relies on the HJB equation. Given a quadratic loss on the control input, the HJB equation can be transformed into a partial-differential equation (PDE) [143]. Dynamic programming and PMP are connected through the viscosity solutions of the aforementioned PDEs [142]. However, in most cases, the HJB equation is hard to solve [144] and does not admit smooth solutions [147]. Most RL based controls [145] rely on optimizing the HJB equation and can be viewed as an approximation of the dynamic programming [146] approach.

In this chapter we follow an alternative approach, where we extend the neural ordinary differential equation framework to solve FC problems. We describe and evaluate the ability of neural ordinary differential equation control (NODEC) to efficiently control non-linear continuous time dynamical systems by calculating FC signals. In Section 2.1, we discuss related work. Section 2.2 summarizes mathematical concepts that are relevant for controlling graph dynamical systems. In Section 2.3, we provide an overview of the basic features of NODEC and formulate conditions for its successful application to solve control problems. In Section 2.4, we showcase the ability of NODEC to efficiently control different graph dynamical systems that are described by coupled ODEs. In particular, we use NODEC to calculate FCs that synchronize coupled oscillators and contain disease dynamics with a limited number of driver nodes. Interestingly, NODEC achieves low energy controls without sacrificing performance. Section 2.5 concludes the chapter.

2.1 Related Work

Previous works used NNs in control applications [148], in particular for parameter estimation of model predictive control [106, 108]. Extensive applications of NNs are also found in the field of Proportional-Integral-Derivative (PID) controllers [148], where the gain factors are calculated via NNs. Shallow NNs have been trained to interact with and control smaller-scale ODE systems [148], without using neural ODEs or deep architectures. Recently, deep NNs have demonstrated high performance in control tasks, and notably on related work on differentiable physics [107] that often use PMP. Deep RL [149] models are also used to calculate control signals and rely on approximations of the HJB equation. Other gradient-based non NN approaches rely on the usage of adjoint methods [150]. Such model approaches are based on the solutions of the PMP principle and calculus of variations solutions. One can also design generic approaches to control network dynamics [151, 152]. Optimal control with NODEC, where the the NN is only a function of time t is extensively studied in Chapter 6. where it is compared with analytically derived methods. The current work focuses on FC methods where the input of the NN is the state vector $\boldsymbol{x}(t)$. We study non-linear dynamical systems, where minimum energy (optimal) controls are not always known. In our work, we always choose state-of-the-art control solutions when available, such as FC [153] and deep RL methods [154, 155], so that we can compare NODEC performance with corresponding baselines. The main contributions of this work are: (i) an adaptive efficient FC approximation methodology with implicit energy regularization properties that relies on neural ODEs, (ii) detailed numerical experiments involving high-dimensional non-linear dynamical systems with minimum driver node constraints, and (iii) an extensively tested codebase that can be easily applied to other nonlinear control applications.

2.2 Feedback Control of Graph Dynamical Systems

A graph $G(V, E)$ is an ordered pair, where V and E $\subseteq$ V $\times$ V are the corresponding sets of nodes and edges, respectively. We denote the number of nodes $|\mathrm{V}| = N$ by N. Although, in network science [156], it is more common to refer to graphs as networks, in this chapter we will use the term "graph" instead of "network" to avoid confusion with NNs. Throughout this chapter, we study dynamical systems on graphs described by the adjacency-matrix $\boldsymbol{A}$, which has non-zero elements $\boldsymbol{A}_{i,j}$ if and only if nodes i and j are connected. We describe controlled graph dynamical systems by ODEs of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{\boldsymbol{A}}\left(t, \boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}(t))\right), \tag{2.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^N$ denotes the state vector and $\boldsymbol{u}(\boldsymbol{x}(t)) \in \mathbb{R}^M$ $(M \leq N)$ an external control signal applied to $M \leq N$ (driver) nodes. The adjacency matrix $\boldsymbol{A}$ in the subscript of $\boldsymbol{f}$ denotes the graph-coupled interactions in the ODE system. For the remainder of the chapter, we omit the subscript as all systems under evaluation are graph-coupled ODEs that have fixed adjacency matrices over time. We use Newton's dot notation for differentiation $\dot{\boldsymbol{x}}(t)$. The function $\boldsymbol{f}$ in Relation (2.1) accounts for

both (time-dependent) interactions between nodes and the influence of external control signals on the evolution of $\boldsymbol{x}(t)$. We assume that the system state $\boldsymbol{x}$ is fully observable. In control theory, the control signal $\boldsymbol{u}(\boldsymbol{x}(t))$ is often calculated via two approaches: (i) by using time as input (i.e., $\boldsymbol{u} = \boldsymbol{u}(t)$) [141] or (ii) by using the system's state at time t as input (i.e., $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}(t))$) [157]. The latter approach is often used in state-feedback control [157], where the control signal is calculated as a function of the difference between the state that the system reached at time t and the control target state $u(\boldsymbol{x}(t) - \boldsymbol{x}^*)$. In this chapter, we focus on state-feedback control and denote control signals by $\boldsymbol{u}(\boldsymbol{x}(t))$. The applicability of the current framework on time-dependent controls is evaluated in detail in Chapter 6.

For the majority of complex dynamical systems, Relation (2.1) cannot be solved analytically. Instead, numerical solvers can be used to calculate approximate solutions of Relation (2.1). As a starting point, one may use an explicit Euler method where for a given state $x(t)$ at time t, the state of the system at time $t + \Delta t$ is $\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + \Delta t\, \boldsymbol{f}\,(t, \boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}(t)))$. Apart from an Euler forward integration scheme, there exist many more numerical methods [158] to solve Relation (2.1). We use the expression $\mathrm{ODESolve}(\boldsymbol{x}(t), t_0, T, \boldsymbol{f}, \boldsymbol{u}(\boldsymbol{x}(t)))$ to indicate a generic ODE solver that uses the right-hand side of Relation (2.1) as an input and computes the state trajectory, or set of state vectors, $\mathrm{X}_{t_0}^{T} = \{\boldsymbol{x}(t)\}_{t_0 \leq t \leq T}$. In Section 2.4, we employ Dormand–Prince and Runge–Kutta schemes as our ODESolve methods.

Driver Node Selection

Our aim is to showcase the ability of NODEC to produce efficient FCs for systems where the number driver nodes approaches the minimum number necessary to achieve control. Thus, we need to identify set of driver nodes that are able to fully control the underlying dynamics. Usually, we are interested in finding the minimum set of driver nodes, which is equivalent to the graph-theoretical problems of maximum matching or minimum edge dominating sets [159, 160]. However, for general graphs, finding the maximum matching set is NP-hard [161, 162]. In our NODEC framework, we determine driver nodes according to two methods: (i) the maximum matching method [129] for disease dynamics and (ii) from stability criteria in the case of Kuramoto oscillators [77]. We denote the set of driver nodes and its cardinality by $\mathrm{B} \subseteq \mathrm{V}$ and M, respectively. Furthermore, we use $\boldsymbol{B} \in \mathbb{R}^{N \times M}$ to denote the driver matrix. The elements $\boldsymbol{B}_{i,m}$ are equal to 1 if node i is a driver node and if the control signal u_m is applied to node i. Otherwise, the driver-matrix elements $\boldsymbol{B}_{i,m}$ are equal to 0. The driver matrix $\boldsymbol{B}$ connects a control input $u_m(\boldsymbol{x}(t))$ associated with a driver node m to the corresponding graph node i for non-zero elements $\boldsymbol{B}_{i,m}$. Although NODEC can be used to evaluate shared and/or interacting control signals [1], in this chapter we evaluate dynamical systems where each control signal $u_m(\boldsymbol{x}(t))$ is assigned to one and only one graph node i; thus only one matrix element $\boldsymbol{B}_{i,m}$ is non-zero per row $\boldsymbol{B}_i$. Literature is rich in studies on driver node placement on graphs, there is considerably fewer work that addresses ways of efficiently finding

[1] In shared control, the same control signal $u_m(\boldsymbol{x}(t))$ is applied to multiple nodes, while in interacting control multiple control signals are applied to the same node i.

control inputs for high-dimensional dynamical systems with a limited number of driver nodes.

Control Energy Constraints

In complex systems, it may not always be possible to apply any control signal to a driver node. Consider a disease that spreads between networked communities (nodes) and a control signal that denotes the intensity of quarantine. Applying a constant control signal with high values indicating blanket lockdown measures may not be acceptable by society. In the given example, our goal would be to contain disease spreading as much as possible, while applying appropriate control signals to the driver nodes. A widely use metric for the intensity of the control signal [129] is the control energy

$$E\left(\boldsymbol{u}(\boldsymbol{x}(t))\right) = \int_{t_0}^{T} \|\boldsymbol{u}(\boldsymbol{x}(t))\|_2^2 \, \mathrm{d}t, \tag{2.2}$$

where $\|\cdot\|_2$ denotes the L2 norm. In our numerical experiments we approximate Relation (2.2) by

$$E\left(T\right) \approx \sum_{k=1}^{K} \|\boldsymbol{u}\left(\boldsymbol{x}(t_0 + k\Delta t)\right)\|_2^2 \, \Delta t. \tag{2.3}$$

In Chapter 6, we show that NODEC approximates optimal (or minimum energy) control signals without the necessity of explicitly accounting for an integrated energy cost in the underlying loss function. Instead, NODEC implicitly minimizes the control energy via the interplay of an induced gradient descent, neural-ODE solver dynamics, and NN initialization. Avoiding the control energy term in a constrained optimization also reduces computational cost of learning compared to solving boundary-value PMP problems [133, 134, 135], or computing solutions of the Hamilton–Jacobi–Bellman (HJB) equation [142, 143, 144, 145, 146]. In the current chapter, we provide evidence that NODEC achieves lower energy and higher performance when compared to different FC baselines for large complex systems.

2.3 Neural ODE Control

As in Section 2.2, we consider a dynamical system (2.1) with initial state $\boldsymbol{x}(t_0)$, reached state $\boldsymbol{x}(T)$, and target state $\boldsymbol{x}^*$. The goal of NODEC is to minimize differences between $\boldsymbol{x}(T)$ and $\boldsymbol{x}^*$ using control inputs $\hat{\boldsymbol{u}}(\boldsymbol{x}(t), \boldsymbol{w})$, where the vector $\boldsymbol{w}$ represents the weights of an underlying NN. We quantify differences between reached and target states with the control loss function $J(\mathrm{X}_{t_0}^T, \boldsymbol{x}^*)$ over the state trajectory $\mathrm{X}_{t_0}^T$. The general NODEC procedure is thus based on finding weights $\boldsymbol{w}$ that minimize a loss function $J(\mathrm{X}_{t_0}^T, \boldsymbol{x}^*)$ under the constraint (2.1), using a gradient descent update over a certain number of epochs. That is,

$$\begin{aligned} &\min_{w} J(\mathrm{X}_{t_0}^T, \boldsymbol{x}^*; \boldsymbol{w}) \\ \text{s.t.} \quad &\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}(t))), \end{aligned} \tag{2.4}$$

where the control signal $\boldsymbol{u}(\boldsymbol{x}(t)) = \hat{\boldsymbol{u}}(\boldsymbol{x}(t); \boldsymbol{w})$ is calculated as an NN output and

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \Delta\boldsymbol{w} \quad \text{with} \quad \Delta\boldsymbol{w} = -\eta \nabla_{\boldsymbol{w}} J(\mathrm{X}_{t_0}^{T}, \boldsymbol{x}^*; \boldsymbol{w}). \tag{2.5}$$

Here, $\eta > 0$ denotes the learning rate. Our proposed method relies on the usage of neural ODEs [136], which are a natural choice for the approximation of continuous-time control signals. Using neural ODEs instead of discrete-time controls allows us to approximate a continuous-time interaction and express the control function $\hat{\boldsymbol{u}}(\boldsymbol{x}(t); \boldsymbol{w})$ as a parameterized NN within an ODE solver (see Figure 2.1).

We show that NODEC can be used to control non-linear graph dynamical systems with different loss functions. It is of particular relevance for continuous time control problems with unknown and intractable optimal control functions. The ability of NODEC to approximate various control functions is established by universal approximation theorems for the approximation of continuous-time control functions with NNs. Similar to RL, NODEC is able to learn control inputs directly from the underlying dynamics in an interactive manner. Contrary to other control approaches [141, 143, 133, 134, 135], we do not impose a control energy constraint directly on our optimization loss function, improving the learning efficiency considerably.[2]

In Algorithms 2.1 and 2.2, we show the two parts of a generic NODEC algorithm that approximates control signals. The main elements of NODEC are: (i) input and target states, (ii) graph coupled dynamics, (iii) NN architecture and initialization, (iv) the parameters of the ODE solver (e.g., step-size), and (v) the gradient descent algorithm and its hyper-parameters, such as the learning rate. Note that Algorithm 2.2 relies on the automatic differentiation methods [163, 164], where the gradients "flow" through the underlying NN that is time-unfolded by ODE solvers [158].

Neural ODE and NODEC Learning Settings

Although NODEC utilizes neural ODEs [136], the learning tasks of both frameworks differ significantly. Neural ODEs model dynamics of the hidden state $\boldsymbol{h}(t)$ of an NN according to

$$\dot{\boldsymbol{h}}(t) = \boldsymbol{g}(t, \boldsymbol{h}(t); \boldsymbol{w}), \tag{2.6}$$

where $\boldsymbol{g}(\boldsymbol{h}(t), t; \boldsymbol{w})$ and $\dot{\boldsymbol{h}}(t)$ represent the NN and hidden-state evolution, respectively. As in Eq. (2.4), the vector $\boldsymbol{w}$ denotes the neural-network weights. Previously, neural ODEs were mainly applied in supervised learning tasks [165] and in normalizing flows [136]. For NODEC, we use an NN as a parameterized function to approximate the control term $\boldsymbol{u}(\boldsymbol{x}(t))$ in graph dynamical systems (2.1). Contrary to supervised applications of neural ODEs [136], our proposed framework numerically solves control problems in an interactive manner, similar to reinforcement learning.

[2] Imposing an energy constraint would require collecting and back-propagating the norm of all control inputs at each time step during training. Using such a back-propagation scheme would increase training times considerably because of the potentially large number of control inputs in large-scale graph dynamical systems.

Algorithm 2.1: A generic algorithm that describes the parameter learning of NODEC.

Result: $\boldsymbol{w}$
Init:: $t_0, \boldsymbol{x}_0, \boldsymbol{w}$, $\boldsymbol{f}(\cdot)$, ODESolve$(\cdot)$, $J(\cdot), \boldsymbol{x}^*$;
Params:: η, epochs;
epoch $\leftarrow 0$;
while epoch $<$ epochs **do**
    // Generate a trajectory based on NODEC.
    $\mathrm{X}_{t_0}^T \leftarrow$ ODESolve$(\boldsymbol{x}_0, t_0, T, \boldsymbol{f}, \hat{\boldsymbol{u}}(\boldsymbol{x}(t); \boldsymbol{w}))$;
    // gradient descent update
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} J(\mathrm{X}_{t_0}^T, \boldsymbol{x}^*)$;
    // or Quasi-Newton with Hessian:
    // $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta H^{-1} \nabla_{\boldsymbol{w}} J(\mathrm{X}_{t_0}^T, \boldsymbol{x}^*)$
end

Algorithm 2.2: A simple ODESolve implementation.

Function *ODESolve*$(\boldsymbol{x}(t_0), t_0, T, \boldsymbol{f}, \hat{\boldsymbol{u}}(\boldsymbol{x}(t); \boldsymbol{w}))$*:*
    // Euler Method
    $t \leftarrow t_0$;
    // State trajectory:a set of state vectors.
    Set $\mathrm{X} \leftarrow \{\boldsymbol{x}(t_0)\}$;
    while $t \leq T$ **do**
        // Computational graph is
        // preserved through time
        // gradients flow through $\boldsymbol{x}$
        $\hat{\boldsymbol{u}} \leftarrow \hat{\boldsymbol{u}}(\boldsymbol{x}(t); \boldsymbol{w})$;
        $\boldsymbol{x} \leftarrow \boldsymbol{x} + \tau \boldsymbol{f}(t, \boldsymbol{x}, \hat{\boldsymbol{u}})$;
        $\mathrm{X} \leftarrow \mathrm{X} \cup \{\boldsymbol{x}\}$
        // Step τ could be adaptive
        $t \leftarrow t + \tau$;
    end
    return X;
end

Learnability of Control with Neural Networks

As reachability of a target state $\boldsymbol{x}^*$ from an initial state $\boldsymbol{x}(t_0)$ implies the existence of a control function $\boldsymbol{u}(\boldsymbol{x}(t))$, we now focus on the ability to approximate (i.e., learn) $\boldsymbol{u}(\boldsymbol{x}(t))$ for reachable target states with an NN.

Proposition 1. *Given that (i) a target state $\boldsymbol{x}^*$ is reachable with continuous time dynamics* (2.1) *and (ii) the control function $\boldsymbol{u}(\boldsymbol{x}(t))$ that reaches the target state $\boldsymbol{x}^*$ is continuous or Lebesque integrable in its domain, then a corresponding universal approximation (UA) theorem applies for an NN that can approximate a control function $\hat{\boldsymbol{u}}(\boldsymbol{x}(t); \boldsymbol{w}) \rightarrow \boldsymbol{u}(\boldsymbol{x}(t))$ by learning parameters $\boldsymbol{w}$.*

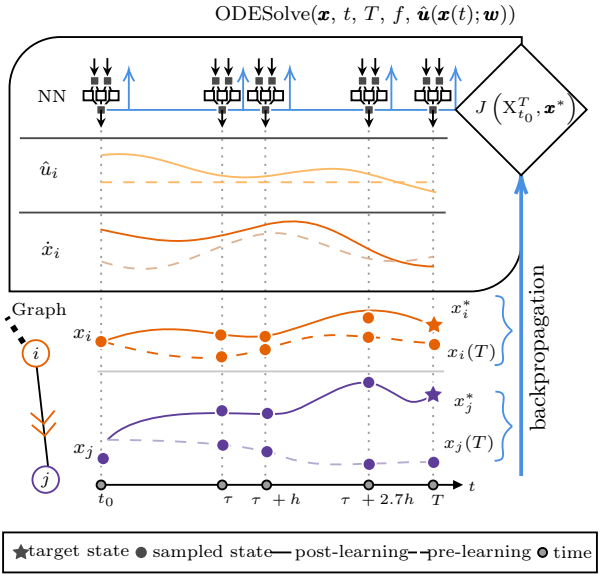


FIGURE 2.1: A schematic that summarizes the training process of NODEC. A NN learns the control within the ODESolve method.

The above proposition holds when both an appropriate UA theorem [166, 167, 168, 169, 170] and reachability [171] requirements are satisfied by the underlying dynamics and the NN controller. Related work indicates UA properties for neural ODEs [172] that can be leveraged to calculate approximate solutions to the control problem Relation (2.4). The ability of an NN to learn control signals has also been covered in the literature outside of the domain of neural ODEs [173, 174, 106, 175]. In the current work, we choose to compare our proposed model to an analytical FC baseline [77] and state-of-the-art RL [155] for non-linear dynamical systems describing Kuramoto oscillators and disease spread.

Learning Loss and Control Goals

To apply NODEC to control tasks, we have to translate a control goal into an adequate learning loss. The choice of the control goal depends on the underlying dynamics, graph structure, and objectives of the control designer. A very common goal in literature [176] is "microscopic" control where each node i has to reach a predetermined state value within time T, i.e. $x_i(T) = x_i^*$. Such a control goal is often applied in industrial applications and may be used to steer electric and mechanical systems towards desired target states [176]. This control goal is achieved by minimizing a metric that quantifies the distance between the target and reached states $\boldsymbol{x}(T)$ and $\boldsymbol{x}^*$. One possible choice of such a metric is the mean squared error

(MSE) $J(\boldsymbol{x}(T), \boldsymbol{x}^*) = \frac{1}{N}\sum_{i=1}^{N}(x_i(T) - {x_i}^*)^2$. When the MSE is used, corresponding optimal control problems may be expressed as convex optimization problems [177]. For more details on the application of NODEC to microscopic loss function, see Chapter 6.

We focus on control goals that do not require a specific target state value for each node, but instead require that constraints over aggregate values or statistical properties of the system's states are satisfied. For the control of certain complex systems, it is useful to consider such "macroscopic" constraints [178, 131]. Often such goals lack exact optimal control solutions, thus offering many opportunities for novel control applications of NODEC.

A common macroscopic control goal is that nodes in the target state are required to be synchronized, i.e. the nodes' states are required to have the same value or constant phase shifts. Such synchronization conditions are often considered in the context of controlling oscillator systems [150, 179]. When synchronizing oscillators, reaching the target state at time T may not satisfy the control goal completely, as we may require the system to preserve the state properties that satisfy the synchronization goal for a time period $[t, T]$. In the current work we showcase that NODEC is able to optimize such control problems.

We also consider more complex control goals, where the system evolution includes coupled ODEs with more than one state variable. In the context of epidemic models, the state $\boldsymbol{x}_i$ of a node i is represented by a vector that consists of multiple state variables. For susceptible-infected-recovered (SIR) models, three state variables, $S_i(t)$, $I_i(t)$, and $R_i(t)$, are used to model the part of a population on node i at time t that is susceptible, infected, and recovered, respectively. A relevant control goal for controlling epidemics is the "flattening" of the curve, or reducing the maximum infected population that occurs at time $t^* \in [t_0, T]$.

Reinforcement Learning

When solving control problems with a dynamic programming approach, the value function is calculated via the HJB equation

$$V_T(\boldsymbol{x}(0), t = 0) = \min_{\boldsymbol{u}(\boldsymbol{x}(t))} \int_{t_0}^{T} J(\boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}(t))dt + J_T(\boldsymbol{x}(T)) \tag{2.7}$$

where we have an objective cost $J(\cdot)$ and a terminal cost $J_T(\cdot)$. As pointed out earlier, the HJB equation is often hard to solve [144] and may not admit smooth solutions [147]. A potential approach would be to use an approximate solution of the value function, which is the main focus of the RL frameworks [145]. NODEC does not solve or approximate the value function to derive the control signal and, thus, deviates from the HJB related methodologies, such as deep RL.

Deep RL is often described as "model-free" and addresses the (i) value function approximation problem and (ii) control problem [145]. A noisy "proxy" signal of the value function, termed as reward function $\rho(\boldsymbol{u}(\boldsymbol{x}(t)), \boldsymbol{x}(t))$ is often used to approximate the value function. In practice the reward is often a designer choice, and different reward functions may be evaluated to determine the one that can be used to

approximate the value function efficiently. We note that RL approaches may suffer from credit assignment challenges, where a reward signal is uninformative regarding the specific actions (especially in terms of time) that help reach the goal [180]. In contrast to RL, the proposed NODEC is not model-free and the underlying gradient descent is directly calculated from the loss function. Therefore, we do not need to consider value prediction and credit assignment. It is possible to design a model-free NODEC by learning the underlying system dynamics with identifier networks [148] simultaneously with control, which could be an interesting future extension of our work. Note that a direct performance comparison between deep RL and NODEC in terms of target loss and convergence time may be considered unfair especially towards RL methods, unless extensive hyper-parameter optimization is performed beforehand.

Although deep RL algorithms may suffer from credit assignment problems and often require higher training samples, their advantage lies on the ability to learn controls of systems that are not directly observable, measurable, or contain noise [145]. When enough training samples are present, it is possible to apply RL to real-world applications related to sustainable decision-making [97], especially when no differentiable models of the underlying dynamics exist, and, thus NODEC cannot be applied. Recent advances in the field of deep RL also show promising results, which can reduce the required training sample sizes and can calculate efficient controls to complex systems [181, 182, 183], especially related to the field of engineering. Nevertheless, it is important to point out the similarity of RL, and especially model-based RL with NODEC, in the essence that both learning methods use NN, gradient based optimization, and can learn similar control tasks. Both methods can be combined across their common technical aspects to enable more complex learning schemes, such as transfer learning discussed in the disease spreading results of Section 2.4 for RL and NODEC baselines that have the same architecture.

2.4 Experimental Evaluation

In this section, we evaluate the ability of NODEC to (i) reach target states efficiently with a limited number of driver nodes, (ii) control different dynamics and losses, and (iii) calculate low energy control signals. We first evaluate the performance of NODEC for two non-linear systems with very different control tasks to showcase its applicability and versatility in computationally challenging settings, for which analytical solutions or approximate control schemes may not exist. We describe the experimental setup by defining the dynamical systems, initial state, control goal, and NN hyper-parameters used for training. The choice of NN hyper-parameters focuses mainly on the network architecture, inputs, optimizers, and training procedures. For the sake of brevity, we omit technical details in the main text and provide further information in the Appendix and in our code [184, 185] and data repositories [186].

Coupled Oscillator Dynamics

Here we study the ability of NODEC to control a network of coupled oscillators via FC, similar to the one used in the study of synchronizing micro-grids [77]. Such systems are used to model power grids [187, 83, 77], especially under the assumption that loss of synchronicity can lead to highly costly black-outs and severe socio-techno-economical disruption [18]. One common control goal for oscillator systems is to reach a fully synchronized target state and stabilize the system over time. This introduces two main challenges: (i) a target state that satisfies this goal needs to be reached and preserved and (ii) the trained model needs to be able to achieve synchronization stability for initial states not seen in training. For continuous time linear time invariant systems and systems that can be linearized, there exist optimal FC methods [188]. Continuous-time oscillatory dynamics may not always be linearizable [77] and exhibit chaotic behavior [189, 190], which cannot be observed in (finite-dimensional) LTI systems. NODEC does not require linearization and could potentially control systems that are costly or intractable to linearize.

In a graph of N coupled oscillators, a possible mathematical description of the evolution of phase x_i of oscillator i with natural frequency ω_i is

$$\dot{x}_i = \omega_i + \sum_m \boldsymbol{B}_{i,m} u_m\left(\boldsymbol{x}(t)\right) + K \sum_j \boldsymbol{A}_{i,j} \hbar(x_j - x_i) \tag{2.8}$$

where $\boldsymbol{A}$ is the interaction matrix, K the coupling constant, and $\hbar$ a 2π-periodic function [77]. By setting $\hbar(x) = \sin(x)$, we recover Kuramoto dynamics [191]

$$\dot{x}_i = \omega_i + \sum_m \boldsymbol{B}_{i,m} u_m\left(\boldsymbol{x}(t)\right) + K \sum_j \boldsymbol{A}_{i,j} \sin(x_j - x_i). \tag{2.9}$$

In accordance with Ref. [77], we consider a target state, in which all oscillators are tightly packed in a synchronized cluster where $|x_j - x_i| \approx 0$ for all i, j. Linearization of the uncontrolled version of Eq. (2.9) within the rotating reference frame yields the target state

$$\boldsymbol{x}^{\diamond} = K^{-1} \boldsymbol{L}^{\dagger} \boldsymbol{\omega}, \tag{2.10}$$

where $\boldsymbol{L}^{\dagger}$ is the pseudo-inverse of the graph Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ and $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_N]$ is the vector of natural frequencies [77]. For the derivation of Relation (2.10), one uses that the mean natural frequency is zero. The quantity $\boldsymbol{D}$ is the degree diagonal matrix, with each element of the diagonal $\boldsymbol{D}_{i,i} = d_i$ being equal to the degree d_i of the corresponding node i.

Learning Loss Function

We measure the degree of synchronization of Kuramoto oscillators in terms of the order parameter [179]

$$r(t) = \frac{1}{N} \sqrt{\sum_{i,j} \cos\left[x_i(t) - x_j(t)\right]} = \frac{1}{N} \sqrt{\sum_{i,j} e^{i(x_i - x_j)}}. \tag{2.11}$$

The control loss may also aggregate the order parameter over time, when the control goals take stability into account. In such a case, one might consider the mean order parameter over time

$$\overline{r}(t) = \frac{1}{T}\int_0^T r(t)\,\mathrm{d}t, \tag{2.12}$$

which approaches zero if the oscillators are incoherent. By discretizing T into Ξ intervals, we can also discretize Relation (2.12) using

$$\overline{r}(t) = \frac{1}{\Xi}\sum_{\xi=0}^{\Xi} r(\xi\tau)\tau, \quad \Xi\tau = T. \tag{2.13}$$

For the numerical calculations we omit τ. Equation (2.13) can be used as a loss function

$$J(\mathrm{X}_\tau^T) = -\overline{r}(t) = -\frac{1}{\Xi}\sum_{\xi=1}^{\Xi} r(\xi\tau) \tag{2.14}$$

to achieve stable synchronization of coupled oscillators. Loss functions that represent periodically and randomly changing oscillator states can be also used as an input for learning control inputs. Instead of using loss function (2.14) that is associated with maximizing the degree of synchronization in the whole network, one may wish to achieve different degrees of synchronization in different time windows, which can be modeled by using time-dependent synchronization losses. Furthermore, forcing parts of a network into different oscillatory states may be possible by minimizing differences between observed and desired oscillator phases in certain subgraphs.

The loss function Relation (2.14) introduces two challenges with respect to the classical MSE loss [177]: (i) it is a macroscopic loss as we do not require to reach a specific state vector $\boldsymbol{x}^*$ to minimize[3] Relation (2.14) and (ii) the loss is calculated over a time interval $[\tau, T]$.[4] In our numerical experiments we observed that using such a loss affects numerical stability, especially for long time intervals, e.g. when $\Xi = 100$ timesteps. Averaging over $r(\xi t)$ may smooth out temporal large variations of $r(t)$, especially for very high values of Ξ. When such drops occur in sampled training trajectories, NODEC learns to achieve high synchronicity only temporarily. NODEC learns controls that yield highly synchronized stable trajectories similar to the FC baselines, when we extend Relation (2.14) by subtracting the minimum order parameter value $\min_{t\in[\tau,T]} r(t)$ over time:

$$J\left(\mathrm{X}_\tau^T\right) = -\left[\overline{r}(t) + \min_{t\in[\tau,T]} r(t)\right]. \tag{2.15}$$

[3] The target states that satisfy this control goal are not unique and not necessarily known, but satisfy $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} r(\boldsymbol{x})$. Since there is no specific dependence on a target state vector, we omit the quantity $\boldsymbol{x}^*$ in the loss function.

[4] The initial time is omitted ($\xi = \{1, \ldots, \Xi\}$ in Relation (2.14)), since we assume that no control is applied prior to reaching the initial state.

Introducing the minimum order parameter term increases the stability of the learned control as the loss creates higher gradients for controls that cause loss of synchronization. We train NODEC on trajectories that may at maximum reach a total time of $T = 40$, and we test and evaluate the performance of the trained NN on random initial states for $T = 150$. As we report in Section 2.4, the described training protocol is able to achieve good performance for these testing parameters.

Control Baselines

A FC baseline for Kuramoto dynamics is presented in Ref. [77]. First, the FC gain vector $\boldsymbol{b}^{(\mathrm{FC})}$ is defined for the control baseline. In accordance with Ref. [77], we use a control gain vector $\boldsymbol{b}^{(\mathrm{FC})}$ instead of a driver matrix. An element $b_i^{(\mathrm{FC})}$ of the gain control vector is assigned to a graph node i and it needs to satisfy

$$b_i^{(\mathrm{FC})} \geq \sum_{j \neq i} \left[|K\boldsymbol{A}_{i,j} \cos(x_i^{\diamond} - x_j^{\diamond}) - \epsilon| - (K\boldsymbol{A}_{i,j} \cos(x_j^{\diamond} - x_i^{\diamond}) - \epsilon) \right]. \tag{2.16}$$

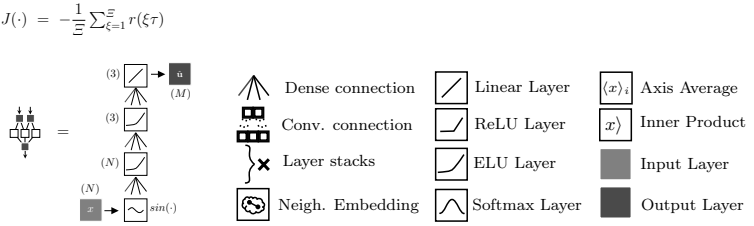
If $\cos(x_i^{\diamond} - x_j^{\diamond}) \geq 0$, the corresponding term in the summation vanishes. We take the equality of the constraint in Relation (2.16) to calculate the control gain vector elements $b_i^{(\mathrm{FC})}$ based on Ref. [77]. The error margin buffer ϵ is implemented as suggested in related work [77] by setting $\epsilon \geq 0$ when selecting driver nodes in Relation (2.16). For $\epsilon = 0$, the driver node selection might be insufficient and it may not be possible to drive the system to a desired target state [77]. Using an error margin buffer increases the driver node selection tolerance and, thus, selects more diver nodes, which can steer the system to a desired target state. The non-zero values of the driver matrix can be chosen arbitrarily, as long as the constraint in Relation (2.16) is satisfied. Non-zero values $b_i^{(\mathrm{FC})} \neq 0$ determine the driver nodes. The baseline control signal u_i for a node i is calculated as

$$u_i(x_i(t)) = \zeta b_i^{(\mathrm{FC})} \sin({x_i}^* - x_i(t)). \tag{2.17}$$

We note that here we follow the notation of Ref. [77] and use a control gain vector $\boldsymbol{b}$ instead of a driver matrix. In driver matrix notation, we iterate over all nodes and select a node i as the m-th driver node by setting the driver matrix element $\boldsymbol{B}_{i,m} = b_i^{(\mathrm{FC})}$ if $b_i^{\mathrm{FC}} \neq 0$. We set the target state in Relation (2.17) to ${x_i}^* = 0$.

We require that FC reaches comparable performance to NODEC in terms of $r(t)$; thus we multiply the vector $\boldsymbol{b}^{(\mathrm{FC})}$ with a positive scalar value[5] $\zeta = 10$. Higher absolute values of $\zeta|b_i^{(\mathrm{FC})}|$ may create control signals that reach the target state in less time at the expense of a higher control energy. As the driver matrix is calculated based on an approximation of the graph Laplacian pseudo-inverse $\boldsymbol{L}^{\dagger}$ of a singular system, optimal control guarantees for minimum energy may not always hold.

[5] We tested several other values before selecting the specific value. Smaller values would lead to a lower degree of synchronization than that achieved by NODEC, but they would require less energy. Higher values would either completely fail to synchronize the system or require very high amounts of energy to achieve similar results.



(a) Kuramoto Controller network.

(b) Legend of neural networks operators.

FIGURE 2.2: NN architecture for controlling Kuramoto oscillators and symbol legend.

Numerical Simulation Parameters

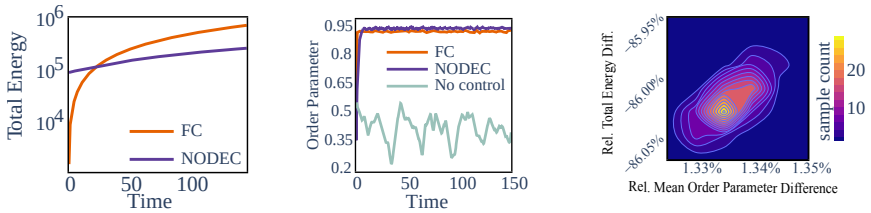
To evaluate the system synchronicity, we calculate the order parameter (see Relation (2.11)), which reaches the maximum value $r(t) = 1$ if all oscillators are fully synchronized.

For our numerical experiments, we create an Erdős–Rényi graph $G(N, p)$ with $N = 1024$ nodes, expected mean degree $\overline{d} = 6$, and link probability $p = \overline{d}/(N-1)$. We generate the driver matrix as in Section 2.4, and select the non-zero elements as driver nodes. To reduce approximation errors due calculation of the Laplacian pseudo-inverse matrix, we set a buffer margin of $\epsilon = 0.1$ in Relation (2.16), when selecting driver nodes. Control signal energy is evaluated with Relation (2.3). Moreover, we set the coupling constant to $K = 0.4$ and sample the natural frequencies ω_i from a uniform distribution $\mathcal{U}(-\sqrt{3}, \sqrt{3})$ [153]. This setting results in approximately 70% of the nodes being assigned as driver nodes.

NODEC Hyperparameters

Only the current system state $\boldsymbol{x}(t)$ is provided as an input for the NN, similar to the baseline described in Section 2.4. We use a fully connected architecture as illustrated in Figure 2.2a. Finally, to calculate the binary driver matrix B for the NN in Relation (2.9), we iterate all nodes of G, and for each node i we assign $\boldsymbol{B}_{i,m} = 1$ and set it as the m-th driver node if $b_i^{(\mathrm{FC})} \neq 0$ and $\boldsymbol{B}_{i,m} = 0$ otherwise. The driver index m increments by one each time a driver node is assigned. We use a binary driver matrix instead of the gain matrix used in FC, as we require the network to learn the control signals per driver node without prior knowledge of the exact control gains.

The current control goal is to stabilize Kuramoto oscillators in a synchronized state over a period of time. The loss of synchronization may occur at any point during this period. To avoid loss of synchronization over a period of time, we train NODEC (see Appendix Algorithm A.1) in a curriculum learning procedure [192], where NODEC is initially trained on trajectories sampled for low values of T. The value of T increases gradually as training proceeds. The learning process in the



(a) Control energy over time.

(b) Order parameter over time.

(c) Relative NODEC performance against FC.

FIGURE 2.3: Comparison of NODEC and FC in terms of energy and synchronization stability.

beginning of the curriculum, when T is very low, allows NODEC to learn controls that steer the oscillators through the transient state between synchronicity and no-synchronicity. As T increases the network also learns controls that preserve the network in the synchronized state.

In FC, the target is often to synchronize the system for different initial states [193]. To train the system for more than one initial state, we use a mini-batch-training procedure that samples 8 random initial states per epoch for training. We observed that randomly sampling an initial state from a uniform distribution in $[0, 2\pi]$ does not improve training performance and fails to learn synchronization. It has been reported in the literature [194] that normally-distributed layer inputs (with zero mean and unit variance) can help NNs converge faster. Therefore, we decided to sample initial states from a normal distribution with zero mean and unit variance. Our results confirm that learning and convergence improve. Sampling initial states enables us to use mini-batches to speed up and stabilize training as well. In the Kuramoto example we use the Adam optimizer [195] for parameter optimization. The complete training scheme is also illustrated in the Appendix Algorithm A.1.

Results

To test the control performance of NODEC, we first sample an unobserved initial state close to the synchronized steady state in accordance with [77]. The initial state values for single sample evaluation (see Figures 2.3a and 2.3b) are uniformly sampled within -10% of the synchronized steady state values, i.e. $x_i \in [0.9x_i^\diamond, x_i^\diamond]$, in order to be close to the synchronized steady state as proposed in Ref. [77]. We observe that the NN achieves a target state with larger order parameter values (see Figure 2.3b) and requires lower energy (see Figure 2.3a) than the FC baseline. We also observe that NODEC requires higher energy and slightly more time to synchronize the system but less to preserve it, compared to the FC baseline (see Figure 2.3b and Appendix Figure A.1).

To determine whether NODEC can achieve synchronization stability regardless of the initial state choice (see Figure 2.3c) and its proximity to the synchronized

steady state, we test the trained model on 100 initial states, with values uniformly sampled in $[0, 1]$. In Figure 2.3c the vertical axis represents the relative total energy difference between NODEC and FC for the same initialization:

$$\left(E_{NODEC}(T) - E_{\mathrm{FC}}(T)\right) / E_{\mathrm{FC}}(T). \tag{2.18}$$

The horizontal axis represents the mean relative order parameter difference calculated as

$$\left(r_{NODEC}(T) - r_{\mathrm{FC}}(T)\right) / r_{\mathrm{FC}}(T). \tag{2.19}$$

NODEC achieves around 1% higher order parameter values and almost 86% less total control energy for all samples. More sophisticated strategies of adapting the constant term ζ in Relation (2.16) could be applied to adapt the driver matrix values in FC. This is, however, out of scope of this thesis. Our results show that NODEC can be adapted to achieve highly synchronized states in Kuramoto dynamics on an Erdős–Rényi graph via FC.

Epidemic Spreading and Targeted Interventions

Designing targeted intervention and immunization strategies [113, 196] is important to contain the spread of epidemics. To study the performance of NODEC in such containment tasks, we will use the susceptible-infected-recovered-type (SIR-type) model [84] that extends the classical SIR model by accounting for quarantine interventions and other preventive or reactive measures for disease containment. In our formulation of SIR-type dynamics, we also account for control inputs and network structure. The "R" compartment in our model is used to describe (i) recovered individuals that were infected and acquired immunity and (ii) removed individuals (i.e., susceptible individuals under quarantine who do not interact with anyone else). The complete state of the epidemic model is now described by a matrix $\boldsymbol{X}(t) \in \mathbb{R}^{4\times N}$, where each row represents the state vector of the corresponding node[6]. The proportion of susceptible, infected, recovered, and quarantined individuals at node i is $\boldsymbol{X}_{1,i} = S_i$, $\boldsymbol{X}_{2,i} = I_i$, $\boldsymbol{X}_{3,i} = R_i$, $\boldsymbol{X}_{4,i} = Y_i$, respectively. The corresponding generalized SIR-type dynamics of node i is described by a set of rate equations:

$$\dot{S}_i(t) = -\beta S_i(t) \sum_j \boldsymbol{A}_{i,j} I_j(t) - \sum_m \boldsymbol{B}_{i,m} u_m(\boldsymbol{x}(t)) S_i(t) \tag{2.20a}$$

$$\dot{I}_i(t) = \beta S_i(t) \sum_j \boldsymbol{A}_{i,j} I_j(t) - \gamma I_i(t) - \sum_m \boldsymbol{B}_{i,m} u_m(\boldsymbol{x}(t)) I_i(t) \tag{2.20b}$$

$$\dot{R}_i(t) = \gamma I_i(t) + \sum_m \boldsymbol{B}_{i,m} u_m(\boldsymbol{x}(t)) S_i(t) \tag{2.20c}$$

$$\dot{Y}_i(t) = \sum_m \boldsymbol{B}_{i,m} u_m(\boldsymbol{x}(t)) I_i(t) \tag{2.20d}$$

[6] We note that here we use capital letters for the SIR-type variables, to follow the common notation in related literature.

subject to the conditions that (i) the total population is conserved and (ii) the control budget is bounded from above by $\mathcal{b}$:

$$\sum_i [S_i + I_i + R_i + Y_i] = N, \tag{2.21a}$$

$$\sum_{m,i} \boldsymbol{B}_{i,m} u_m(\boldsymbol{x}(t)) \leq \mathcal{b}. \tag{2.21b}$$

The driver nodes $B_{i,m} = 1$ can be selected via different methods, e.g. the nodes or communities that are willing to apply proactive and reactive measures. In our simulations, driver nodes are selected with the maximum matching method [159, 160]. Furthermore, we assume that the epidemic originates from a localized part in the graph and we minimize the proposed epidemic loss in Relation (2.22) for a different part of the graph G^*. The initial spread and target sub-graph are illustrated Figure 2.4. The parameters β and γ are the infection and recovery rates, and $u_m(\boldsymbol{x}(t))$ describes the effect of containment interventions (e.g., quarantine, mask-usage and distancing). When an NN controller (NODEC or RL) is used, we set $u_m(\boldsymbol{x}(t)) = \hat{u}_m(\boldsymbol{x}(t))$. These terms are used to model preventive and reactive measures, respectively. For example, susceptible individuals may isolate themselves and completely avoid infection ($S \to R$) until the pandemic passes (preventive) or infected individuals are quarantined and put to intensive care to avoid spreading and recover ($I \to Y$) (reactive measure).

The described control problem is complicated by several factors. First, the budget constraint [see Relation (2.21b)] does not allow assignment of high control values across all nodes. Second, we need to distribute limited intervention resources dynamically across structurally similar nodes. Unlike in networks with community structure, where isolating single nodes can effectively control epidemic spreading, the regularity of the square lattice does not admit such a control approach.

Another possible formulation of time-dependent control targets in epidemic modeling is to weigh the number of new infections with a discount factor to prioritize minimizing current infections over minimizing future infections. For network epidemic models, degree-based approximations can be used in conjunction with optimal control theory to derive interventions for such loss functions. For more information, see e.g. Ref. [197].

Learning Loss Function

The control goal is to "flatten" the curve, i.e. to delay and minimize the mean infected fraction over nodes in the subgraph G^*, which has no overlap with the part of the graph containing the initial spreading seed. Based on these control goals, we formulate the following loss function:

$$J(\mathrm{X}_{t_0}^T, \boldsymbol{X}^*) = [\max_{t_0 \leq t \leq T} \bar{I}_{G^*}(t)]^2, \tag{2.22}$$

where $\bar{I}_{G^*}$ denotes the mean fraction of infected individuals in G^*. This control goal is macroscopic, as we do not know the exact feasible state $\boldsymbol{X}^*$ for which

$I^*(t^*) = \arg\min_{I(t)} J(I(t))$ that minimizes such a loss. Furthermore, the exact time t^* at which the minimum loss is achieved is not known, and therefore we need to evaluate samples from the state trajectory $\mathrm{X}_{t_0}^{T}$ to determine t^*. Similar to Relation (2.15), the current control goal requires loss calculations over a time interval. Moreover, this loss is not calculated over the whole state matrix $\boldsymbol{X}$ but only on the infected state I_{G^*} of the target subgraph. Intuitively, one would trivially achieve the proposed goal if there are no further constraints. If nodes that connect the subgraph G^* to the rest of the graph cannot be controlled efficiently, then achieving the control goal becomes non-trivial. Tackling the outlined epidemic control problem allows us to evaluate NODEC on a complex control task (see Section 2.4) with applications in disease control.

Control Baselines

A baseline that takes structural node properties (e.g., node degree or centrality) into the account, may be a good baseline for graphs with structural-heterogeneity, but not for regular structures like lattices. Clearly, a weak baseline is random control (RND), where we assign random control inputs to driver nodes with $u_m(t) = \mathcal{b} c_m / \sum_{m'=0}^{M} c_{m'}, c_m \sim \mathcal{U}(0,1)$. However, a targeted constant control baseline (TCC), which in the presence of an "oracle" assigns constant control inputs $u_m(t) = \mathcal{b}/M$ to every driver node in G^*, is a strong baseline for constant control. As TCC is a static control, it already protects the driver nodes from $t = 0$ on, so TCC-controlled nodes will be infected very slowly. Assigning all budget to all driver nodes of interest also minimizes wasted "containment" budget. Still, distributing more budget to a smaller number of nodes increases the L2 norm of the control, making controls very expensive when considering quadratic energy costs. To have a control with less energy, it is important to distribute the budget to more nodes, therefore enabling more global containment and less constant containment on the target sub-graph.

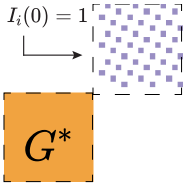


FIGURE 2.4: Intial infection and target subgraph, illustrated on the lattice graph.

We also study the performance of neural dynamic control baselines, such as continuous-action RL, with fully-connected NNs or our variant (see Figure 2.5) as policy architectures (see Sections A.2 and 2.4). Only one of the three evaluated training routines of RL provided high-performance results. We tested: SAC [154], TD3 [155], and A2C [198], but we report only the results of TD3 which were more competitive with respect to NODEC. To allow RL to tackle the SIR-type control problem, we first implement SIR-type dynamics as an RL environment. The input of RL is the tensor of all SIR-type states at time t. We consider an observation space, which includes continuous values in $[0, 1]$ and has dimension $4 \times N$. RL actions $a_m(t) \in \mathbb{R}$ are continuous values for each driver node and correspond to control signals. Once the actions are passed to the environment, a pre-processing operation takes place to convert the RL action into valid control signals (see decision network of Figure 2.5). RL is allowed to provide change the control

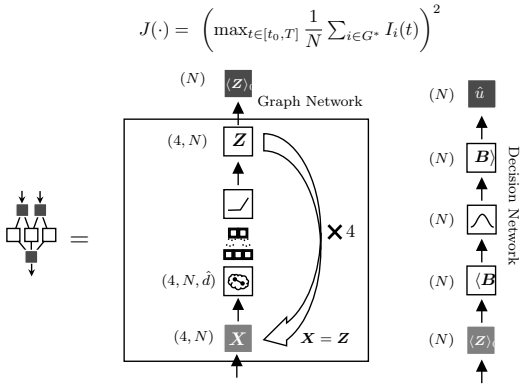


FIGURE 2.5: NODEC architecture for controlling SIR-type dynamics.

signals to (interact with) the environment in a fixed discrete time interaction interval $\Delta t = 10^{-2}$ during training. Lower interaction intervals were also considered, but required longer training and did not seem to improve performance. For RL, we need to express the control goal as a reward function which is used for the approximation of action value function within the RL framework. We tested several reward designs and we describe this process in the Appendix Section A.2, but we simulate best performance with the following reward function:

$$\rho(t) = \begin{cases} 0, & \text{if } \bar{I}_{G^*}(t) \leq \max_{\tau<t}(\bar{I}_{G^*}(\tau)) \\ -\bar{I}^2_{G^*}(t) + (\max_{\tau<t} \bar{I}_{G^*}(\tau))^2, & \text{otherwise} \end{cases} . \tag{2.23}$$

Numerical Simulation

To determine the target time T, we observe the SIR-type dynamics ($\beta = 6$ and $\gamma = 1.8$) on a 32×32 lattice without control and set its value to the time at which the mean infection over all nodes is approximately zero. Initially, the epidemic starts from a deterministic selection of nodes in the upper-right quadrant. For all control strategies, the budget (maximal number of control interventions) is $\mathcal{b} = 600$. Given that RL takes considerably longer to converge and that we were required to perform a much more extensive hyper parameter search, we showcase our experiments only on the lattice graph and a single initial state. Our control goal is to contain epidemic outbreaks (i.e., "flattening" the infection curve) in the sub-graph G^*, which is located in the bottom-left quadrant (see Figure 2.7). All baselines are compared with timestep $\Delta t = 10^{-3}$.

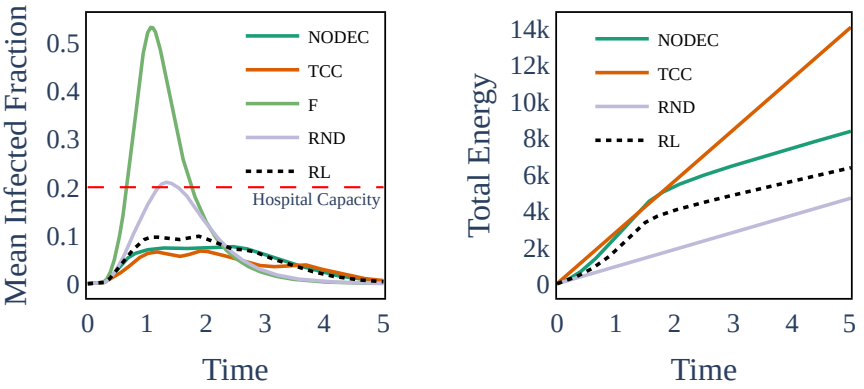
NODEC Hyperparameters

From a technical perspective, the SIR-type dynamics introduce extra state variables. Therefore, fully-connected layers will require one to estimate considerably more parameters. We observe that neither NODEC nor RL converged to a high-performance solution when using fully-connected layers, and we thus omit these results. Furthermore, the control task requires the network to optimize a loss that is not calculated over whole graph, but rather on a specific sub-graph. NODEC has no direct information on which nodes are part of sub-graph G^*. The information is provided via the minimization of the learning loss-function in Relation (2.22). Back-propagation happens at time $t^* = \arg\max_{t \leq T} J(I_{G^*}(t))$. This time is approximated by preserving a sample of states when using the ODESolve, and picking the maximum observed peak infection from that sample.

As the existing neural architectures discussed in Section 2.4 did not perform well, we switch to an architecture that includes the graph structure. To leverage the information of the graph structure and generate efficient control signals that "flatten" the curve, we decide to design a more specialized NN architecture that includes the information of the graph-structure within its layers. For that reason we use a Graph NN (GNN) architecture (see Figure 2.5 and Section A.2). The same GNN architecture is implemented in the RL baselines as the policy network. GNNs calculate hidden features by aggregating node features and the graph structure as inductive biases [199, 200]. Following Relation (2.20), we observe that disease spread happens between direct neighbors in the graph, and, thus, such inductive biases can potentially benefit from taking the graph structure into account. Furthermore, the resulting models can converge faster and require less computational effort to reach better learning performance, as the inclusion of the graph structure in the NN can reduce considerably the number of learned parameters [199], especially for large graphs. For example, a single layer fully connected network that takes as input one state variable per node would require $N \times M$ parameters to be learned for N nodes and M driver nodes, without accounting for bias parameters. Using a graph convolutional NN and a pooling layer or an aggregation layer as an output layer to reduce the output of the convolutional layer to a vector of M control values, requires as many parameters as the desired kernel size and hidden channel size of the convolutional layer. The graph convolutional layer is expected to have a considerably smaller number of parameters compared to a fully connected architecture for a large graph.

For our experiments, we use a learning rate $\eta = 0.07$ and the Adam optimizer. The same GNN architecture is implemented in the RL baselines as the policy network. GNN encountered fewer numerical instabilities during training and allowed for efficient learning without curriculum procedures. We use a training procedure for SIR-type control as shown in Appendix Algorithm A.2 that preserves the best performing model in terms of loss.

The hidden state matrix $\boldsymbol{Z}$ is calculated from the GNN and then provided as an input to the decision NN (see Figure 2.5 right side). The decision network contains operations that enforce the budget and driver constraints by applying a softmax activation function and calculating control signal outputs for the driver nodes. The



(a) Infected fraction.

(b) Total energy.

FIGURE 2.6: SIR-type control evaluation. NODEC versus baselines: reinforcement learning (RL), targeted constant control (TCC), random constant control (RND), and free dynamics with no control (F).

decision network contains no learned parameters and is included inside the NODEC architecture and RL environment. Transfer learning [201] between NODEC and RL can be achieved by pre-training the GNN network with NODEC and then using it as an RL policy. RL achieves the same performance as NODEC when transfer learning is tested. Further fine tuning of the pre-trained policy with RL does not improve performance of NODEC in this setting, but transfer learning indicates a possible future extension of combining model-based training with real-world model-free fine tuning.

Results

Our main results are summarized in Figure 2.6 and Table 2.1 and indicate similar superior performance of TCC and NODEC compared to the other control strategies, but with lower energy costs for NODEC. In Figure 2.6, we observe that NODEC is providing strong protection with total energy costs that are not as high as TCC (see Table 2.1). If we assume that the proposed system will reach maximum hospital capacity at 20% of the infected fraction in the target sub-graph, we observe that TCC, RL, and NODEC are sustainable control strategies. In Figure 2.6a and Table 2.1, NODEC underperforms TCC with approximately 1% higher maximum infection fraction, but requires almost 41% less control energy. The effectiveness of the control can be attributed to the adaptive nature of NODEC. The other adaptive baseline, RL requires around 54% less energy than TCC but allows for 2.1% higher peak infection compared to NODEC. The effectiveness of targeted adaptive controls in time can be used to model and examine the effectiveness of proposed real-world

long-term pandemic control strategies, such as rolling lockdowns [202] and/or vaccine allocation [203, 197].

NODEC achieves better performance at the cost of higher energy compared to RL. We note that RL approaches may suffer from credit assignment challenges, where a reward signal is uninformative regarding the specific actions (especially in terms of time) that help reach the goal [180]. However, even after testing different reward designs and parameters settings, no RL framework managed to perform better than our baselines. It may be possible that extensive reward engineering, and other model upgrades may lead to a better performance. In contrast to RL, the proposed NODEC is not model-free and the underlying gradient descent is directly calculated from the loss function. Therefore, we do not need to consider value prediction and credit assignment. It is possible to design a model-free NODEC by learning the underlying system dynamics simultaneously with control similar to Ref. [107], which could be an interesting future extension of our work.

TABLE 2.1: Total energy $E(T)$ and peak infection $\max_t(\bar{I}(t))$ achieved by different epidemic spreading control methods.

Control	Peak Infection	Total Energy
TCC	0.068	14062.6
NODEC	0.078	8356.6
RL	0.099	6358.0
RND	0.210	4688.9
F	0.532	0.0

The spread of the epidemic, target subgraph, and controls of the main baselines are illustrated in Figure 2.7. RL and NODEC calculate control signals that change over time and slowly fade out as $t \to T$. We also observe that controls persist in some driver nodes even then the infection wave is over (see also Appendix Figure A.3). This behavior is also observed in other baselines that satisfy the equality of the constraint Relation (2.21b) (RND and TCC). The budget constraint Relation (2.21b) allows control signals to sum up to the budget value $\mathcal{b}$. The implemented NN architecture calculates controls by multiplying the budget with a softmax activation function output over a hidden state output from the learned GNN architecture (see Figure 2.5 right side). The output of the softmax activation function is non-zero by definition[7] and thus the NN always calculates non-zero control signals. Once the infection wave has traversed the graph, both RL and NODEC controllers spread the control over several nodes, thus decreasing required control energy[8]. This outcome

[7] In practice control signals may approach 0 due to floating point errors.

[8] Looking at the control energy Relation (2.2), we observe that low absolute value control signals assigned over many driver nodes may produce lower energy values compared to very high absolute value control signals applied to fewer driver nodes.

is an artifact of the softmax activation function, but it may also indicate the implicit energy regularization properties of NODEC. On the contrary, the higher energy costs of TCC keep increasing, as high control signals remain in place after the infection wave has passed.

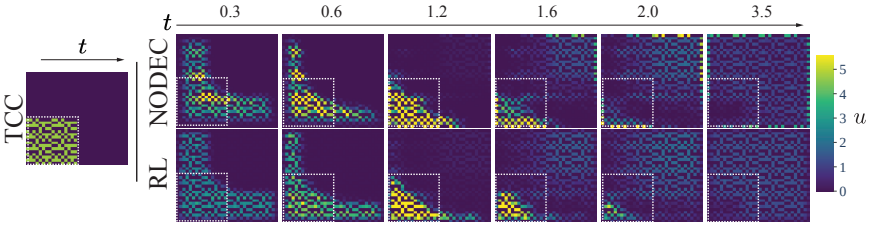


Figure 2.7: Control trajectories for SIR-type dynamics. Colorscale plots represent 99.5% of the presented values for dynamics with NODEC controls.

2.5 Discussion and Conclusion

Neural ODE control approximates dynamical systems based on observations of the system-state evolution and determines control inputs according to pre-defined target states. Contrary to Ref. [136] that parameterizes the derivative of hidden states using NNs, our neural-ODE systems describe controlled dynamical systems on graphs. In general, NNs are able to approximate any control input as long as they satisfy corresponding universal approximation theorems. Such control methods can assist subsequent decision-making [39] by approximating optimal solutions to sustainability problems related to infrastructure [77, 83, 84]. The study continuous time models under optimal control assumptions with NODEC can become useful and provide outputs that can assist organizational decision-making, especially when studying potential disruption scenarios in managed infrastructures.

However, in practice, NODEC needs to deal with different numerical hurdles such as large losses and stiffness problems of the underlying ODE systems. By testing NODEC on various graph structures and dynamical systems, we provide evidence that neural ODE-based control approaches are useful in FC and that numerical hurdles can be overcome with appropriate choices of both hyperparameters and ODE solvers. Nevertheless, modelling socio-technical and environmental systems as ODEs requires specific assumptions to be met. In real-world, the underlying systems might not be always observable, continuous in time, and also might contain considerable amounts of noise. Therefore, modelling such systems with ODEs might not be possible or contain errors arising from design bias and violation of assumptions. In that case, control approaches such as NODEC may not be able to calculate efficient or realistic controls. On the contrary, model-free methods, such as deep RL might be a more preferable method to tackle such control challenges. It is important that

the control designer and the decision-makers are aware of system characteristics and determine whether the usage of NODEC or other control techniques is appropriate to assist decision-making.

From a value-sensitive design perspective, current developments in deep learning have provided us with many useful tools that are used to simulate, analyze, and predict a wide spectrum of social dynamics [204]. Still, the amount, granularity, and quality of data required to perform such studies often conflict with privacy, fair treatment, and autonomy of individuals [51]. When designing experiments that involve any deep learning method, and especially NODEC, privacy preservation and autonomy should be of the highest priority. For example, in our simulations, the selection of driver nodes is done with assumption that the nodes that satisfy the minimum driver node constraints are willing to accept the lockdown, social distancing, and mask mandate measures. This should not be the case in a corresponding real-world application, especially in social systems where nodes are usually matched to individuals. In case of social studies, frameworks of participatory and self-determined control can be developed instead, where individuals are free to choose whether they want to be selected as driver nodes. NODEC is shown to control dynamics, even when a fraction of available driver nodes is randomly selected, e.g. as shown in Appendix Figure E.6, indicating that it could potentially approximate participatory controls.

Implicit energy minimization properties of NODEC, which are further studied in Chapter 6, can prove useful for the study of social systems. Individuals and societies may not always accept any imposed high impact controls in their lifestyles, as they may prove oppressive and authoritative, e.g. long term strict lock-downs. In that case, controls that have minimum impact over time (or lower energy) may be more acceptable, e.g. dynamic lock-downs that are not very prohibitive for most citizen's activities. Concluding, calculating low energy controls that sustain operational infrastructures and are acceptable by societies may be crucial for sustaining operational democratic societies.

Chapter

3

Value-Sensitive Design for Decentralized Sustainable Product Ratings

Creating more sustainable consumption patterns turns out to be imperative for mitigating climate change and supporting a more viable future of our society [205, 206]. Food systems, in particular, play a key role, influencing 12 out of the 17 Sustainable Development Goals of the United Nations [207]. Food supply is already leading on total greenhouse gas emissions [208, 209], while food demand is expected to grow by 50% along with an increase of the global population from 7 to 9.8 billion people by 2050 [210]. Thus, introducing policies that promote more sustainable consumption patterns is critical. Yet, it has proven to be insufficient to overcome the "attitude-behavior gap" [40]. For instance, choosing grocery products according to a broad spectrum of (often conflicting [211, 212, 53]) sustainability criteria requires the processing of an overwhelming amount of information and, as a result, the price often remains the dominant choice criterion. Such information is often summarized and its processing is automated, i.e. by product labels or smart phone shopping assistants that scan product barcodes [213, 214, 215]. Ensuring a value-sensitive design in terms of accountability, transparency, privacy, self-determination, and an overall practicality by a seamless integration into the shopping process, remains a grand challenge. But, designing digital shopping assistants, e.g. smart phone apps, for such values turns out to be a requirement for consumer trust (see IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems [216]). Given the broad scope and complexity of the subject of sustainability and the co-determination of consumer decisions by price [217], product choices based on sustainability criteria can be particularly prone/vulnerable to manipulative nudging when personalized decision-support systems use personal data.

Centralized approaches, as the one shown in Chapter 2, may not be applicable in scenarios were individual freedoms and preferences are compromised by strict control. Individuals have the right to pursue their own morals and social norms, thus, centralized controls that allow for technocratic and autocratic implementations might endanger such right [51]. In this chapter we show how value-sensitive design can empower a bottom-up shift to more sustainable consumption. To make this possible, we built a novel and general-purpose shopping assistant [218] and tested it at two real-world supermarkets in Estonia (Retailer A [219]) and Austria (Retailer B [220]). The shopping assistant was implemented as an Android smart phone application

(app) for decision-support: It rates products (as well as product categories) that are in front of the consumer in a shop according to *personal sustainability criteria* (preferences) and in a privacy-preserving way. These preferences cover the categories of *environment*, *social*, *health*, and *(product) quality* so that consumers can improve their health while supporting socially favorable production conditions and making choices with better environmental impact. Products that better match consumers' preferences are rated close to 10 and those that oppose the preferences receive values close to 0. The consumer can interact with the app to change sustainability preferences using a continuous Likert-scale slider in the range $[0, 10]$. Ratings are explainable to consumers by making transparent why certain products receive higher ratings than others.

The value-sensitive design of this privacy-preserving digital shopping assistant is distinguished from other mainstream approaches (see Figure 3.1d) by integrating the following values in decision-support: (i) Adding environmental, social, health, and quality criteria in consumers' decisions. (ii) Limiting manipulative nudging. (iii) Self-determination in consumers' preference choices. (iv) Privacy preservation. (v) Explainability. (vi) Practicality and compatibility with the existing shopping process.

These values are realized by a design process that tackles the challenges to provide (i) *transparent and explainable personalized product ratings*, (ii) *a seamless integration into the consumers' shopping process*, and (iii) *scalable, high quality product information on sustainability*. The scope and combination of inter-disciplinary methods required for this is quite exceptional, covering a new value-sensitive approach, its implementation within a sophisticated system, and the testing of this approach in real-world settings using rigorous social science methods. This sets the results of this chapter apart from most other state-of-the-art studies (see Table B.14 in the Appendix).

Transparent and explainable personalized product rating. The first challenge of value-sensitive design is addressed by designing a novel content-based recommender algorithm for the personalized rating of products that is decentralized and privacy-preserving. In contrast to user-based collaborative filtering algorithms that rely on the collection of sensitive historical consumer data to compare purchase profiles [221], the novelty of our algorithm is that personalization is taking place on the smart phone and as a result, the sustainability preferences of a consumer remain localized on the consumer's device, i.e. private. This is possible via a scalable, distributed communication protocol that handles a consumer's request for public summarized product information on sustainability (sustainability indices). When this information is localized on the smart phone, efficient calculations turn the sustainability index of products into personalized product ratings. The implications of our value-oriented and preference-based design approach are significant: without sharing sensitive personal data with third parties, manipulative nudging that serves corporate interests and may oppose personal preferences or sustainable consumption patterns is limited [222]. As a result, consumers can trust that the rating of the products is a result of their own intrinsic values expressed via their sustainability preferences. Moreover, the calculation of a product rating is accountable to the

consumer, who can visually explore which product information and preferences are mainly responsible for the overall rating (see Figure B.13 in the Appendix). The explainability is also localized on the smart phone so that no third parties can manipulate the perception of a consumer on why certain products receive higher or lower ratings.

Seamless integration into the consumer shopping process. For the second challenge, a seamless integration in the shopping process is crucial for the practicality and adoption of the solution in retailer shops. Barcode scanning makes the comparison of different products cumbersome [223]. Instead, in our system, consumers automatically view all nearby products of a category, e.g. all different pasta products, on their smart phone and can therefore make efficiently an augmented comparison while moving along the shelves (see Figure 3.1 and Figure B.12 in the Appendix). This novel augmented reality experience is made possible via a low-power bluetooth beacon localization technology that has been deployed and extensively tested on retailers' site [224]. Consumer localization in the shop does not require a centralized collection of GPS traces which tends to be unreliable, privacy-intrusive and not suitable for indoor environments [225].

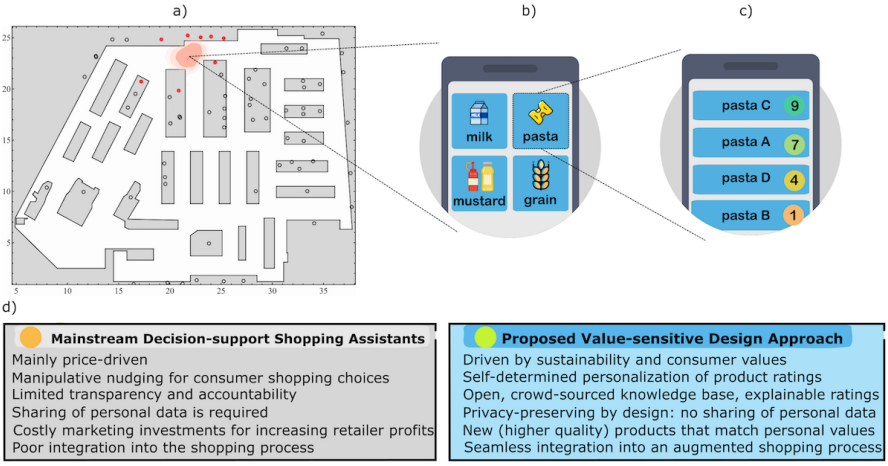


FIGURE 3.1: **An outline and comparison of the proposed value-sensitive, preference-based decision-support process**. (a) A consumer's localization at Retailer A. The red dots on the map denote the product categories in close proximity. (b) Presenting the product categories in close proximity to the consumer. (c) Product ratings close to 10 denote better matching of the product to the consumer's sustainability preferences. (d) Comparison of the proposed value-oriented, preference-based design approach with mainstream decision-support shopping assistants.

Scale and quality of product information on sustainability. This third challenge is addressed by designing and populating an open knowledge base. It is based on a new sustainability ontology with which a transparent and accountable

reasoning for the personalized rating of products is performed. Ontologies have been influential and have often accelerated scientific progress in biology and beyond [226, 227, 228]. They systematically dissect and structure complex concepts to reason and develop a shared understanding that motivates their use in this work. The designed knowledge base consists of information from retailers, online data sources with domain and crowd-sourced knowledge, domain experts, who supported this project during its lifetime, as well as the wisdom of crowds [229, 230], for example, by crowd-sourcing the analysis of Wikipedia data (see Table B.16 in the Appendix). The knowledge base is actually a set of 795 associations between 15472 products and 25 consumer preferences belonging to the four sustainability categories of environment, health, social and quality. See Table B.1-B.7 of the Appendix for a complete list of all preferences and how they are obtained. Formally, the association between a product keyword (e.g. meat) and a preference keyword (e.g. vegetarian) is quantified in the range $[-1, 1]$, with -1 representing opposition (meat does not fit vegetarian diet) and 1 representing support (lettuce fits vegetarian diet). In-between values represent, for instance, the relative positive effect of different vitamins or the relative negative effect of different additives/preservatives in several health indicators. Therefore, the design of the knowledge base is the selection of such keywords (tags) according to formal semantics as well as the reasoning about their in-between association score (196 product and 61 preference tags as shown in Figure B.4 of the Appendix). This semantic information is used to calculate a non-personalized *sustainability index* of a product for certain sustainability preferences by aggregating the association scores among the assigned product and preference tags. Moreover, rebound effects that model the incompatibility of Sustainable Development Goals [211, 212, 53] at the level of consumer choices can be measured at the design phase by analyzing the semantic links that interconnect products, preferences and their tags.

Field tests. We conducted novel and ambitious field tests at two supermarkets from May to November 2018. These go beyond earlier survey questionnaires [231, 232, 233], lab experiments [234] or gamification [235]. We implemented the actual value-sensitive and preference-based decision-support system to empower consumers to shift their shopping choices to sustainable consumption. As a result of the actual system implementation, a more realistic assessment of the shopping shift can be made. A total number of 323 (Retailer A) and 69 (Retailer B) participants with a loyalty card were recruited by the retailers and project staff members on site by offering coupon discounts of up to 100 euros. The loyalty card provides access to baseline historical purchases allowing us to compare consumers using the app (treatment condition) with similar consumers not using the app (no treatment). The rewards are designed to incentivize a regular shopping behavior, limit dropouts and biases. Consumers using the app fill in an entry survey, choose their (baseline) preferences that can be changed later and start shopping with the app. For the purpose of this study, the app usage and purchases are traced but remain anonymized. After the end of the test period consumers collect their coupons by answering an exit survey.

Outline of analysis. The data collected during the field tests allows us to analyze whether consumers shift their shopping choices to other products that better respect their own values, self-determined via their sustainability preferences. It

turns out that consumers are usually ready to pay more for higher quality products, which are healthier and more sustainable [217]. We dissect the complex link of sustainability preferences with price by analyzing whether such a link originates from consumer preferences or from biases in the knowledge base we developed. We further identify the profiles of consumers' preferences and illustrate to what degree they are adjusted to meet their expectations of how products should be rated.

3.1 Methods

Here we illustrate more details of the field tests and the causal impact analysis as well as the localization solution at retailer shops. We also outline the ontology design, the product rating methodology, the preference profiles and preference interactions.

Field Tests and Causal Impact Analysis

Recruitment. Each shopping visit at retailer shops is rewarded with 10 euros. By analyzing the history of purchases, 10 visits are expected on average during the field tests and therefore the total maximum amount of 100 euros incentivizes and rewards regular shopping behavior. The study has received ethical approval by the ethical committee of ETH Zurich. To minimize biases originated by different human explanations of how to use the smart phone app (see Figure B.11 in Appendix), an integrated tutorial requires completion before consumers start using the smart phone app. A FAQ based on questions that arose during living lab experiments before the field tests was also included.

Causal impact analysis The Dynamic Time Warping (DTW) [236] algorithm is used to compare the time series data of consumers that used the smart phone app (treatment consumer) against all consumers that did not use the smart phone app (non-treatment consumers) with purchase records up to 13 months before the field tests (excluding the last month as a buffer zone). Each comparison results in the similarity ranking of consumers, who did not use the smart phone app, from which the top-k nearest neighbors are selected to form the group for assessing sustainable consumption. Since different values of k yield different groups for comparison, the k is selected via a dynamic time warping of weekly expenditure before the treatment as shown in Table B.20 of the Appendix.

The temporal distance between consumers is measured over the three matching criteria (behavior covariates). More specifically, the Euclidean distance with dynamic time warping is used to determine the top-k nearest neighbors for the deflated monthly total budget spent and the sustainability index of the average purchased products per month. In contrast, the Jensen-Shannon distance (divergence) [237] with dynamic time warping is used for the distribution of deflated monthly budget spent per product category as such distance metric is known for measuring the similarity of distributions. For each consumer that used the smart phone app (treatment consumer), we rank all other consumers that did not use the app (non-treatment consumers) according to the three distance metrics. Since there are three distance metrics and we cannot determine which one has higher influence, the average ranking

of the consumers that did not use the app is calculated over all combinations of distance metrics.

Localization of Products

A privacy-preserving indoor localization system is designed for the two retailer shops [224]. It derives the list of products that are in close proximity to consumers. It relies on Bluetooth low energy beacons installed on the shop floor of both retailers. The beacons are aligned with the shop map to provide absolute anchors to the shop coordinates. By analyzing the signal strength of the beacons using triangulation and a map logic on feasible positions and consumer movements, the position of the consumers in the shop is estimated, see Figure 3.1a. This consumer position is related to the coordinates of product groups in close proximity. The rating of these products is calculated and presented to consumers' smart phone as shown in Figure 3.1c. The localization system is privacy-preserving as all calculations are performed at the consumer's smart phone. Moreover, beacons do not obtain any information from consumers and therefore retailers cannot derive consumer movements from this localization system. The deployed technology uses off-the-shelf hardware which yields low hardware costs. The obtained precision has an inaccuracy of about 2 meters. The granularity of product group positions ranged from several meters down to zero accounting the intrinsic ambiguity of product groups positions on top of each other.

Ontology Design

Primitive concepts, tags and rules. The ontology is designed to quantify the association between products and preferences, e.g. to what extent a certain product is for a vegetarian diet, fair trade etc. To measure such associations, we introduce a common alphabet of characteristics for products and preferences. This alphabet is a set of *keywords (tags)* that represent *primitive concepts*. They form the semantic space and scope of sustainability. Subsets of these primitive concepts compose a vocabulary of product and preference tags, while a primitive concept is not further decomposed to keep the ontology practical and consistent [238] within its scope, i.e. a primitive concept is regarded disjoint within a chosen scope of sustainability. For example, the preference tag "vegan" can be composed by the two primitive concepts "production with no animals" and "production with no animal products". Moreover, products and preference tags are assigned to products and preferences respectively based on logical rules. For instance, to assign the product tag "low fat" to a product, a logical rule could determine the number of grams in fat, relative to the total product weight, contained in the product and/or whether the product has a low-fat label. Over 600 such rules are created for this purpose using the Drools framework [239]. We focused on food products. They are the ones that populate the knowledge base and connect to product tags. In summary, the design of the ontology consists of (i) the choice of the primitive concepts, (ii) their assembly to product and preference tags and (iii) the creation of logical rules to connect the

tags with products and preferences. These actions required domain knowledge from experts (WWF, Greenpeace, Ethical Consumer, VKI), reliable data sources (e.g. EU reports) as well as the wisdom of the crowd by running the Social Impact Data Hack [240] to mine and structure information from Wikipedia, for instance, branding information (see Section B.3 in the Appendix).

Association scores. The association between a product and a preference tag is measured by their shared primitive concepts that satisfy a preference tag. We distinguish between positive and negative associations by determining for each pair of product and preference tags subsets of primitive concepts that semantically support or oppose the preference tag, i.e. the preference tag "vegan" supports the primitive concept "no animals involved in production" but opposes the concept "animal product". Therefore, the association score comes with positive and negative values in the range $[-1, 1]$ by summing up the associations between supported and opposed primitive concepts (see Section B.3 of the Appendix for more details).

Reduction design principle. The construction of product and preference tags should adhere to the *reduction design principle*: (i) Between two tags with the same primitive concepts, one and only one should be assigned to a product or preference. (ii) When two tags assigned to a product or preference share primitive concepts, these primitive concepts should be removed and form a new tag. In the example of Figure 3.2, the reduction design principle is violated if the product tag AC is assigned to the product, or, the preference tag BC is assigned to the preference. We prove in Section B.3 of the Appendix how this principle minimizes the error of overlapping tags when the association scores are aggregated to calculate the rating of a product. Violations of the reduction design principle may result in excessive influence of certain preferences on the product rating. In practice, these artifacts may be captured by consumers, whose adjustments of preferences provide additional countermeasures against the error of semantically overlapping tags.

Experts' guideline. We propose a high-level guideline to populate the sustainability knowledge base according to the proposed ontology. This guideline can be used by domain experts to guide the construction process and is outlined as follows:

1. Identify relevant primitive concepts based on (i) the product categories (i) the available product data and (iii) the scope of sustainability preferences (goals).
2. Create product and preference tags using the primitive concepts such that these tags represent how product/preference characteristics oppose or support a product/preference.
3. Create rules that connect the product tags with products, and the preference tags with preferences.
4. Apply the reduction design principle between all combinations of product tags and preference tags that have overlapping concepts.
5. Calculate association scores between product-preference tag pairs in the range $[-1, 1]$.
6. If necessary, go back to Step 1, add or remove primitive concepts and repeat the process.

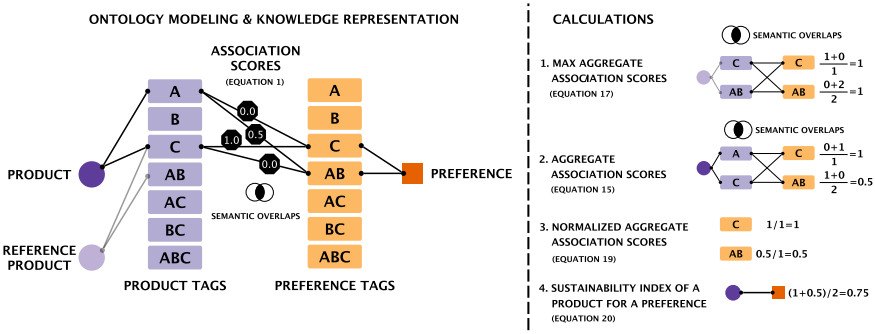


FIGURE 3.2: **Calculation of the sustainability index using the ontology of products and preferences**. Assume an alphabet of primitive sustainability concepts represented in this simple example by $\{A, B, C\}$. Combining the primitive concepts results is the word vocabulary $\{A, B, C, AB, AC, BC, ABC\}$ of product and preference tags. Using rules based on experts' knowledge and verified data such as ingredients of products, the product tags A and C are assigned to a product. Similarly, a sustainability preference is designed by a composition of the two preference tags C and AB. We can now calculate the association scores between the product and preference tags in an automated way (without expert knowledge) by taking the intersection $\cap$ of the tags sets as follows: $|\{A\} \cap \{C\}|/|\{C\}| = 0$, $|\{C\} \cap \{C\}|/|\{C\}| = 1$, $|\{A\} \cap \{AB\}|/|\{AB\}| = 0.5$, $|\{C\} \cap \{AB\}|/|\{AB\}| = 0$. The sustainability index of a product for a given preference is calculated by the average normalized aggregate association scores of the assigned preference tags as demonstrated in this numerical example. The equations refer to the Appendix.

In Step 1, experts determine the scope of the sustainability by defining primitive concepts that capture key characteristics of products and preferences. Experts need to be aware of the products for which they design the ontology, the available information they have about these products as well as the sustainability preferences that should be captured. In Step 2 they can start combining these concepts into tags with the purpose of representing support or opposition to product characteristics and preferences. In Step 3, experts can assign these tags to product and preferences and can formalize rules under which these assignments are made. The processes of the first three steps are the most tedious ones and requires knowledge, experience and a good overview of the available information. As an example of facilitating such processes, we performed workshops with several stakeholders during the project lifetime and organized the Social Impact Data Hack [240]. Step 4 applies the reduction design principle to improve the consistency of the ontology. Step 5 performs the calculations of the association scores based on the (automated) calculations illustrated in Figure 3.2. In practice, these calculations are often calibrated by experts to reason about the association scores based on ground truth knowledge. For instance, consider a study that shows evidence about the effect of different preservatives on health. Obviously, the cause of such effects may be related to chemical or biological phenomena at a very low granularity level that is not captured

within the scope of the designed sustainability ontology. In this case, association scores measuring can be calibrated to reflect the relative effect of preservatives according to the findings of such a study. Finally, the process can repeat by adding or removing primitive concepts. The motivation for this iterative process is to better capture the whole range of preferences, decompose further primitive concepts to make the ontology more granular, add/remove rules, expand product categories or enrich the knowledge based with new datasets. During the ontology design, we performed over 10 iterations for validation purposes and the quality criterion for convergence was how well the product rating could be justified to consumers during the preliminary living lab tests.

Product Rating: Sustainability Index and Preferences

Sustainability index. It quantifies the support or opposition of a preference by a set of product characteristics found in a product. This support or opposition is compared to a product, existing or hypothetical ('reference product' in Figure 3.2), that has all possible characteristics that can support or oppose respectively a preference. Figure 3.2 illustrates the involved calculations. The sustainability index between a product and a preference is measured using the normalized association scores aggregated over the connected product and preference tags. The normalized aggregated association score of a product-preference pair is the normalized aggregated association score of a product averaged over all preference tags assigned to the preference (Calculation 4 in Figure 3.2). Each normalized aggregated association score between a product and a preference tag (Calculation 3 in Figure 3.2) is calculated by the aggregated association scores of the product tags assigned to this product (Calculation 2 in Figure 3.2) divided by the maximum association score between the reference product and the preference tags of the preference (Calculation 1 in Figure 3.2).

Insights on sustainability of production. Note that by calculating for each preference the density of the sustainability index over all products, new opportunities arise to reason about the following: (i) The sustainability profile of different retailers. (ii) New ways (preferences) with which producers can improve products with a more sustainable profile. (iii) Market gaps where new business ecosystems can evolve with stronger involvement of producers and consumers to accelerate sustainable consumption. For instance, the densities in Figure B.10 of the Appendix confirms the more sustainable profile of Retailer B products across the preferences, e.g. higher sustainability index for animal rights, fair trade, recyclability and green farming. Improvements can be made by either introducing new products with better sustainability footprint over these preferences or by improving the existing production practices of the available products.

Product rating. Note that the sustainability index does not require any personal information for its calculation. It only relies on the information of the sustainability ontology, i.e. primitive concepts and tags, that we make available as public-good knowledge. As such, it can be calculated in public computational infrastructure, i.e. servers, public clouds, etc. In contrast, the calculation of the prod-

uct rating requires personalization with consumers' preference choices that remain by design locally on their smart phones to protect privacy and limit manipulative nudging. As a result, the product rating is calculated on consumers' smart phones using the sustainability index values retrieved remotely using a distributed protocol of message passing between smart phones and a project server. The calculation is performed on-demand by consumers when they navigate in the retailer shop and request the rating of the products that are in their close proximity. For each product, the rating algorithm multiplies the sustainability index with the degree of opposition or support of each preference, measured by the distance (offset) from the median preference score (5, remaining neutral). These calculations are summed up and divided by the sum of all distances from the median preference scores. Section B.3 in the Appendix outlines in more detail the product rating calculation and its computational complexity. The (unscaled) product rating calculation is summarized as follows:

$$\text{product rating} = \frac{\text{sum of all (sustainability indices} * \text{preference offsets)}}{\text{sum of absolute preference offsets}}, \quad (3.1)$$

where the product rating values can be scaled to match different grading systems of different countries ($[0, 10]$ in the field tests as supported in the Section B.3 of the Appendix).

Explainability Two levels of rating explainability are provided to consumers: (i) *product tags* and (ii) *preferences*. Consumers can learn about how each product characteristic influences the rating value by solving Equation B.32 of Appendix for a certain product tag, given that all other variables are known. Similarly, consumers can know how each offset of their preferences contributes to the product rating by solving Equation B.31 of the Appendix for a certain preference offset.

Preferences selection. The selection of preferences was made on the basis of providing a broad spectrum of different sustainability indicators with which consumers can express their preferences. However, this spectrum is not too broad to the extent of creating a cognitive overload for consumers and lack of comprehension about which preferences influence the rating of products and why. This is critical for the effectiveness of the rating explainability. Moreover, a lower number of preferences decreases the computational cost of the rating algorithm and improves the usability of the smart phone app (see Section B.3 of the Appendix). This balance is a result of the following process: (i) Participation of several stakeholders in the ASSET project meetings and workshops providing insights about how grocery product choices influence different sustainability criteria. (ii) Preliminary living lab experiments and smaller-scale field tests for feedback acquisition on the preferences. (iii) Choice based on available data, i.e. product and preference tags. Preferences with very similar preference tags or very few preference tags are removed or merged. The set of the final preferences shows a balance between several individual criteria on health (13) versus collective criteria in environment, social and quality aspects (12). The consumers' feedback during the preliminary tests also determined the *strict preferences* of gluten-free, lactose-free, vegan and vegetarian products. Fully

supporting such preferences results in excluding products that do not fully satisfy them even though they may satisfy other (non-strict) consumer preferences. In other words, strict preferences cancel association scores with other preferences.

Preference Profiles and Interactions

Preference profiles. The Ward hierarchical clustering approach [241], with which the preference profiles are extracted, is selected based on its interpretable clusters and high scoring in the following evaluation metrics [242]: silhouette, Dunn index, Davies-Bouldin and Calinski-Harabaz. It uses Euclidean distance given that preference scores are continuous, not sparse and different preference scores are comparable, with the exception of strict preferences that may introduce biases. No normalization or standardization is performed given the same scale among preferences.

Interactions of preferences The sustainability preferences show correlations that originate from the (i) anticipated association between different product/preference tags (ontology-level) and (ii) the actual linking of products to product tags (product-level). This information at the ontology and product-level respectively can be used to reason about potential rebound effects [53]: undermining a sustainability preference by satisfying another one.

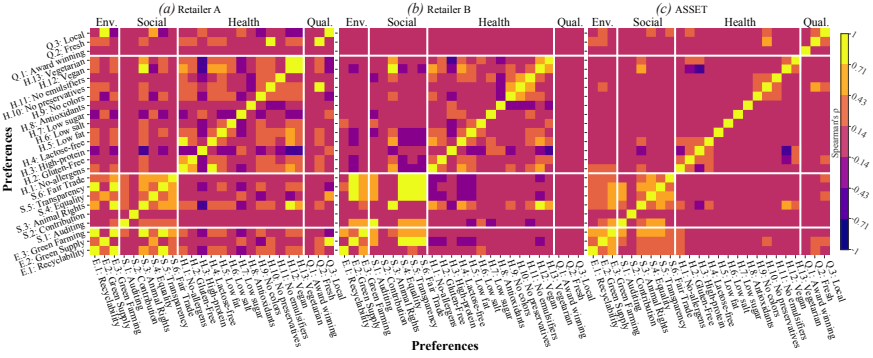


FIGURE 3.3: **Interaction effects between different sustainability preferences**. Ontology-level: Correlation of association scores for (a) Retailer A and (b) Retailer B. Product-level: (c) Correlation of sustainability indices for a common set of products between Retailer A and Retailer B . Positively correlated preference pairs across the x and y axes indicate dependence whereas negatively correlated preference pairs indicate potential rebound effects.

1. **Ontology-level**: The interaction of two preferences is measured as follows: (i) Find the shared product tags between the two preferences. (ii) For each of the two preferences and shared product tag, calculate the average association score among the preference tags of the preference and the shared product tag. (iii)

Calculate the Spearman correlation coefficient of the average association scores among the two preferences.

2. **Product-level**: The interaction of two preference at the product-level is measured as follows: (i) Calculate the sustainability index for each product-preference pair. (ii) Calculate the Spearman correlation coefficient of the sustainability indices among the two preferences.

These Spearman correlations are shown in Figure 3.3. Potential rebound effects are observed between health/quality preferences vs. environmental ones, while a highly rated product may be caused by multiple environmental preferences that share several common product tags.

3.2 Results

We first demonstrate the shift of consumer behavior towards more sustainable consumption, meaning products that are rated higher than 5. The sustainable shopping behavior is measured using the weekly expenditures made for such highly ranked products, normalized to $[0, 1]$, using the observed maximum value. The expenditure values are deflated using the Harmonized Index of Consumer Prices [243]. The consumers that did not use the app (control group) were matched with consumers that did use the app. The following matching criteria (behavior covariates) were applied: (i) The deflated monthly total budget spent, (ii) the distribution of deflated monthly budget spent per product category and (iii) the sustainability index of the average purchased products per month. Matching reduces the control group from 3438 to 532 for Retailer A and from 1843 to 59 for Retailer B. The total consumers who participated in the field tests are filtered out to 148 (Retailer A) and 30 (Retailer B) consumers (treatment) by keeping the consumers who have seen the rating of the products before they have purchased it. The weekly expenditures of the consumers using the app are predicted based on "historic" data in order to estimate the shopping behavior of the treatment group in case they did not use the shopping app. This is done via causal impact analysis [89] using spike-and-slab causal inference via structured Bayesian times series [236]. The predicted values are compared to the actual weekly expenditures made when using the app (see Figure 3.4, where the prediction horizon covers the entire field test period).

Our results confirm a statistically significant increase of expenditures for highly rated products: (i) The absolute effect is about 20% for each retailer. (ii) The relative effect is 36.7% for Retailer A and 41% for Retailer B. The cumulative effect is depicted in order to show how this shift towards expenditures for more sustainable products unfolds over the period of the field test. Table B.22 in the Appendix summarizes the results of the causal impact analysis.

A few additional observations can strengthen the conclusions of Figure 3.4: 68% and 75% of the total purchased products in Retailer A and Retailer B, respectively, are highly rated (> 5) products that are seen in the app. Novel purchases of (i) products purchased for the first time and (ii) products whose rating is seen in the app also increase, namely, by 22% for Retailer A and and 16% for Retailer B.

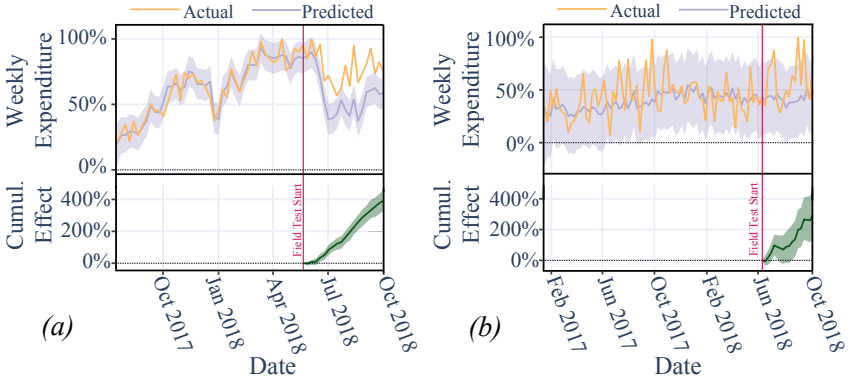


FIGURE 3.4: **Shift to more sustainable consumption during the field test period**. Actual vs. predicted normalized weekly expenditures made for highly rated products (> 5). (a) Retailer A and (b) Retailer B. We find a high statistical significance of $p < 0.001$ for a significance level of $\alpha = 0.01$.

The survey results of Figure B.5, B.6 and B.7 of the Appendix furthermore confirm the following: (i) High product ratings reflect consumers' preferences. (ii) New products are discovered via the products rating. (iii) The rating of products appears justified to consumers. (iv) Consumers are more aware of sustainability aspects after using the app.

Note that the two retailer shops have different weekly expenditure patterns. Retailer A exhibits stronger seasonality effects than Retailer B, which considerably relies on customers with a student profile. Customers of Retailer B seem to have a more diverse demographic profile and a stronger interest in sustainability.

Relation Between Sustainability and Price

During the field test, 16% of the field test consumers at both retailers pay at least 10 euro-cents more on average for each novel product by supporting at least one of the three environmental preferences (see Table B.1 in Appendix). This number is significant given that the price level of most products is low. Moreover, the percentage of consumers who regularly interact with the smart phone app (the ones in the center of our causal impact analysis increases over time to 39% for Retailer A and 30% for Retailer B.

We next assess whether the purchased products rated high within each product category are also the more expensive ones. First, both product deflated prices and the product ratings are normalized (see Section B.4 of Appendix for details). Figure 3.5 shows the frequency of the rated products that were viewed and purchased against price and rating. Retailer A clearly has a higher number of products with high prices and ratings as reflected by the high values on the top right of the heatmap. Similarly in Retailer B, for each normalized price value per product unit, a higher

number of products are viewed and purchased that have higher rating values. We further measure the non-linear relation between product price and product ratings with the Spearman's correlation coefficient across product categories. We observe a positive correlation coefficient of $\rho = 0.14$ and $\rho = 0.19$ for Retailer A ($p = 0.01$) and Retailer B ($p = 0.02$) respectively. This suggests that highly rated products may come with slightly higher prices.

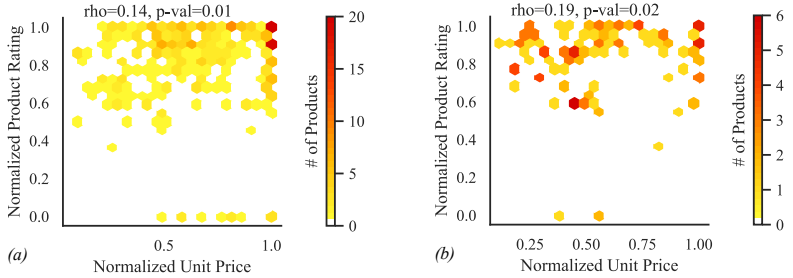


FIGURE 3.5: **Highly rated products correlate to some extent with increased prices.** (a) Retailer A and (b) Retailer B. Normalized product unit price vs. normalized product rating for the frequency of purchased products, whose rating is seen in the app. Frequency values are derived at the level of product categories. Note the increased frequency of purchased products with higher price as the product rating increases. This is particularly visible for Retailer A with a higher number of data records.

The question that arises here is whether this positive correlation stems from the consumers' personalization, i.e. their choices regarding sustainability preferences, or the actual knowledge base and ontology design that is universal for all consumers. To answer this question we repeat these measurements, but instead of the normalized product ratings we used the min-max normalized sustainability index for each preference. The Spearman's correlation coefficient between the sustainability index of products and product prices per category is calculated for each preference across product categories. Out of 50 correlation values evaluated at each retailer, only one correlation value is > 0.5 and statistically significant. This confirms that no price biases are observed in the designed ontology in favor of more expensive products. Nevertheless, it is possible that certain preferences result in a sustainability index that is positively or negatively correlated with price due to the nature of this preference. For instance vegetarian diet products may come with lower prices only because they do not contain meat that is usually expensive. See Tables B.26, B.28 and Figure B.9 in the Appendix for an evaluation of all preferences.

Sustainability preference profiles

To understand how different consumers influence the product rating via personalization, we extract preference profiles by clustering consumers based on the preference choices they made during the field tests. We distinguish between *initial preferences* and *preference adjustments*. The initial preference choices are the very first ones

made before consumers are exposed to the product rating functionality. They reflect the intrinsic intentions of consumers towards the sustainability aspects. The preference adjustments are made throughout the field test after the initial preferences were set.

Initial preferences. The profiles of the initial preferences are extracted by clustering consumers using the Ward hierarchical clustering approach [241]. Seven clusters are calculated for each retailer as shown in Figure 3.6a. Note that most of the profiles consist of choices with a significantly high score in "environment" and "(product) quality" as well as a high score in the category "social". The scores for "health" shows a significant variation among the different profiles that reflects the diversity of dietary needs: (i) Two neutral profiles with scores around 5 are observed for both retailers. (ii) There is a profile that penalizes strict preferences such as vegan, gluten/lactose-free, etc, with low score. (iii) The majority of the profiles retain a high score for health indicators such as no preservatives/colors/emulsifiers, etc. The majority of consumers avoids radical scores in health and expresses their sustainability profile via preferences regarding environment and quality aspects.

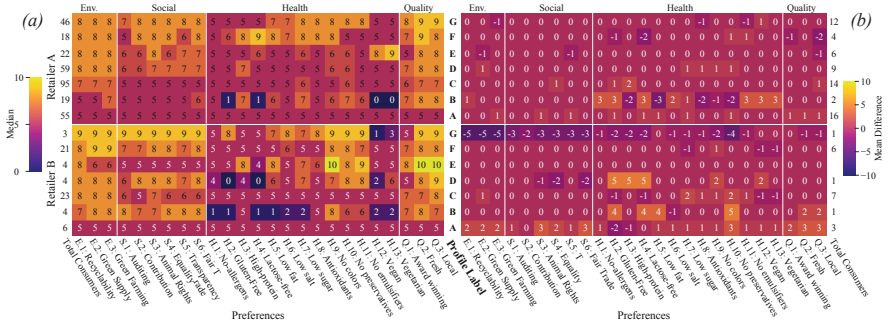


FIGURE 3.6: **Preference score profiles over the four sustainability aspects for Retailer A (top) and Retailer B (bottom)**. Capital letters are labels relating to the extracted preference profiles. They are the same in both figures. The numbers on the left and on the right of each figure refer to the number of consumers in each profile. (a) Initial preference profiles demonstrating the high preference scores in environment and quality aspects. (b) Minimal targeted adjustments made during the field test confirming that product ratings match the expectations expressed via the sustainability preferences.

Preference adjustments. We assume that consumers adjust their preferences if they are not satisfied with how products are rated or if they aspire to explore new products. Consumers may also try to match the rated products with the ones they use to buy. We measure the mean difference between the initial preferences and all adjusted preferences per profile to understand how different consumer groups adjust their preferences to meet their expectations. Figure 3.6b shows the profile adjustments. We observe the following: (i) The majority of preference scores is on average unaffected, indicating that consumers usually do not need to adjust their preferences and they preserve their initial intrinsic sustainability preferences. This

also validates the ontology from the consumers' perspective. (ii) Consumers with initially neutral preferences increase their preference scores, especially in Retailer B. (iii) Consumers who initially maximize the preference scores slightly roll back to more moderate preference values, especially at Retailer B. Apparently this rollback is related with a consumers' openness towards experiencing radically different highly rated products or with the ontology design and rebound effects that results in buying unexpected products. Similarly, the group of consumers that initially penalizes health preferences with low scores makes adjustments to higher scores.

3.3 Discussion and Outlook

The findings of this chapter have several implications for consumers, retailers and producers. They also have an impact on science, policy and at an institutional level. We outline here the challenges and the new pathways that this chapter opens up.

Consumers. We show that a value-sensitive, preference-based design empowers more informed, transparent and accountable shopping choices tailored to personal sustainability values. As a result, consumers explore and finally buy new products. Compared to related work limited to health or environmental indicators for sustainability, we study a broader, yet finite set of such indicators (preferences). In the future, consumers themselves may expand or even limit this set according to their intrinsic priorities. Augmented shopping processes with smart phones may still disrupt regular shopping routines for some consumers. It is unclear to what extent and scale consumers can adopt such technologies at this moment. However, this was also the case with online shopping a few years ago and nowadays efforts such as Amazon Go [244] showcase the feasibility of such technologies in the market. The applicability of the value-sensitive shopping assistant in the context of online shopping is promising and may simplify its use by consumers. In terms of consumers' trust on products rating, new models are required to preserve a satisfactory level of explainability in the long run. This is especially the case when such models need to capture an expanding spectrum of product characteristics as well as the dependencies and incompatibility of sustainable development goals [53]. The transparency and accountability of the knowledge base can be further enhanced with blockchain solutions as a means to securely verify sustainable production practices as well as the traceability of supply chains [245, 246]. Recent studies also confirm how indispensable the active consumers' involvement is to evolve a viable and accurate knowledge base on sustainability via citizen science initiatives [206, 229].

Retailers. Retailers can pioneer future consumption patterns by offering products that are ethically more aligned to consumers' sustainability preferences. They can focus on selling a higher number of highly rated products that might come with slightly higher prices (or profit margins), while they serve consumers' values on sustainability.

Producers. Value-sensitive design provides new opportunities for producers to offer more attractive, higher quality products to retail markets, by capturing (i) consumers' sustainability preferences and (ii) reducing the sustainability gap of existing products. Moreover, producers have new opportunities to sell higher quality

products at a higher price. This incentivizes the improvement of production practices, considering consumer preferences regarding environment, health or worker rights. In other words, thanks to welcomed recommendations of higher-quality products, more sustainable production can pay off.

Science. By demonstrating the shift of individuals to sustainable consumption, we also open up new opportunities for social coordination: moving from individual sustainability preferences to collective actions, which meets sustainable development goals in a bottom-up way. Scientific methods from the areas of game theory, mechanism design, incentive design, socio-technical optimization and learning [247, 248, 249, 70, 30, 26] are becoming applicable to further support sustainability movements in society. Methods such as the ones of this chapter and a transdisciplinary scientific approach are required to improve and accelerate the development of sustainability knowledge bases, using responsible AI and the wisdom of crowds [250, 251].

Policy making and institutions. We foresee new opportunities for environmental non-profit organizations to educate and interact with the general public by publishing their own personalization templates for consumers, for instance, adopting the WWF or Greenpeace priorities for sustainability. Such customizable personalization is supported by the built system. Legislation could protect the open and public good character of a knowledge base for sustainability. Such a knowledge base should not be used as an unquestionable universal ground truth for sustainability. Instead, we envision the active, global involvement of various stakeholders such as public organizations, scientists and especially citizens to capture and develop the evolving nature of sustainability values. Promising related initiatives include the European Commission Environmental Footprint initiative or the Global Product Classification standardizing institution GS1. In other fields such as biology, initiatives for ontologies and knowledge bases have had significant impact, for instance the Gene Ontology Consortium [227]. Overall, the EU project ASSET [218], in the context of which the research proposed in this chapter has been carried out, is a promising blueprint showing how sustainable consumption might be scaled up in a participatory way.

Challenges. We identify several challenges that arise when designing and implementing sustainable recommender systems. The main challenge lies with correctness of data and related interpretations. There are vast amounts of data available in different forms, such as organizational reports [20], research articles [11], and retailer databases. Combining and structuring data across different sources can prove extremely challenging. Crowd-sourced data are also used, but may contain inaccuracies and minor errors, especially if the data providers are adversarial or not interested in providing accurate inputs. Such challenges can be addressed by iterative and repeated evaluations of the system, in which experts and the crowd identify data errors and attempt to fix them. Finally, the proposed expert system may contain designer bias, especially in the design of the proposed ontology and recommendation algorithm. "Nudging" may be reduced by the introduction of self-determined preferences, but still preferences are pre-selected by experts and may shift the focus of individuals towards a subset of sustainability challenges that require resolution. Allowing individuals and experts to introduce, modify, and

decide preferences and relevant product characteristics through time could reduce the “nudging” effect further.

Chapter

4

From Decentralized to Hybrid Basket Recommendations

Consumption is one of the main driving factors of industrial production, and thus has great environmental and societal impact [252]. This impact is the aggregated individual impact of everyday consumer choices. Every day, consumers are urged to take several decisions to fulfill their personal goals and values, mainly originating from individual needs and wishes. Shopping decisions often aim to satisfy several criteria, such as the taste, the ability of purchased products to be combined together, available consumer budget, and health implications of consuming the product. Trade-offs arise when optimizing for personal values, e.g. when a consumer needs to decide between a healthier and cheaper product. Such trade-offs are mainly addressed by multi-objective optimization in modern recommender systems [253], and often involve a high number of objectives to be satisfied. Value-sensitive design can be applied to create systems that support individual decisions when resolving these trade-offs in Chapter 3. Following a value-sensitive design indicates that personal values can be incorporated in the system design, often introducing new constraints and objectives to the optimization. Thus, combining value-sensitive design and sustainable decision-making results in even more complex and challenging trade-off optimization problems.

Supporting value-sensitive sustainable decisions has been evaluated in Chapter 3, often in a qualitative and/or non-optimal manner. Self-determined systems that are based on retailer data, crowd-sourcing, and expert input, have already been successfully tested in the past. Although the effect of sustainable recommendations on individual purchases has been confirmed in Chapter 3 and related literature [254], there is little quantitative analysis on the collective impact of more sustainable decisions on emissions, pollution, resource usage, and personal values. Furthermore, most existing analytical models do not deal or represent real-world data on individual/microscopic level, but rather consists mostly of theoretical economical and climate models that evaluate sustainability on a macroscopic or societal level [100, 102]. A welfare objective function that includes analytical terms representing satisfaction from consumption and environmental pollution is often optimized [29, 102]. The optimization outcomes are then used in analysis and policy making. Although macroscopic and/or centralized models, such as the ones evaluated in Chapter 2, can provide useful insights, it may not be possible to anticipate for errors in estimation of

objective functions and system state. Criticism on centralized models often focuses on poor practical applications and potential high estimation errors [255], as well as potential challenges to addressing personal values and morals [51]. Data-oriented approaches that focus on individual/microscopic level can be directly applied in real-world scenarios in the form of mobile applications and website recommender systems as shown in Chapter 3. Relevant recommender systems and analyses mainly focus on diets or single product recommendations [256, 40]. Finally, hybrid approaches that combine centralized and decentralized models are seemingly sparse in literature.

In this chapter we consider the value-sensitive recommendation for sustainable baskets as a multi-objective optimization problem, which aims to improve an intended basket for purchase. A potential solution approach would be to model the system as a mixed integer programming problem (MIP) and then try to provide a solution via existing optimization techniques. Sustainable basket selection problems may be modelled as complex multi-dimensional non-linear optimization problems. The problem of selecting the optimal number of products that fill a consumer basket under budget and value-sensitive constraints may highly resemble a multi-objective and multi-dimensional knapsack problem [257]. Solving such problem with constraint optimization and linear/dynamic programming may prove challenging, especially as it may not admit efficient Polynomial time approximations [258] and linear relaxation methods may not yield desired solutions [257]. Another widely applied approach using regularization techniques can also be considered to solve such problems and would require the fitting and interpretation of Lagrange coefficients of each objective [259]. The current chapter focuses mostly on the usage of evolutionary algorithms, such as Multi-Objective Natural Evolution Strategies (MO-NES) [260] and Reference point Non-dominated Sorting Genetic Algorithm II [261] (RNSGA-II), which are shown to work on multi-objective optimization problems that have multiple (> 5) objectives. Furthermore, a novel genetic algorithm, termed Gradient Guided Genetic Algorithm (G3A) is introduced, to determine whether incorporating NNs and available product features in the genetic optimization procedure, such as environmental impact quantities and product price, helps finding optimal baskets for recommendation.

The main contributions of the current chapter are to: (i) propose and formalize a new real-world multi-objective optimization problem for recommendations of personalized sustainable baskets, (ii) create and analyze a novel synthetic dataset based on sustainability and consumer real-world data, (iii) propose effective existing baselines for this problem, and (iv) design and implement a novel deep learning framework for mixed integer programming multi-objective optimization problems, as well as (v) evaluate a hybrid approach that combines a decentralized system with information from a centralized system. In Section 4.1 an explicit formalization of the proposed optimization problem is presented. Therein constraints, trade-offs, and objectives originating from value-sensitive design and SDGs are combined to form a multi-objective optimization problem. Appendix Section C.5 summarizes the creation and analysis of the synthetic dataset, which combines consumer transactions and product data from "The Complete Journey Dataset" by the Dunnhumby grocery store [91], emission and resource usage indicators from Ref. [11], and nutrition

information from Food Agricultural Organization Food Balance Sheets [262]. The proposed dataset structure can be used to generate similar real-world datasets by online retailers to enable value-sensitive sustainable recommendations. Section 4.2 introduces a novel deep learning architecture, termed gradient guided genetic algorithm (G3A), which combines the ability of evolutionary algorithms to solve complex multi-objective problems and the ability of NODEC to efficiently and timely control complex processes, such as genetic evolution. Section 4.3 presents the results of experimental evaluation of G3A and other multi-objective optimization algorithms on the problem discussed in Section 4.1. Section 4.4 allows for centralized input when recommending G3A solutions in a privacy preserving manner, based on Chapter 5. Section 4.5 conclused the chapter.

All relevant data and code for the current chapter are reported in Sections C.1 and C.2 and will be available online. Further technical details, such as hyperparameters, training time, and baseline comparison are found in the Appendix Sections C.4 and C.5.

4.1 Personalized Sustainable Baskets

In the current chapter, value-sensitive sustainable recommendations are formalized as a multi-objective optimization problem of selecting combinations of discrete quantities over $N = 132$ distinct products. First, an intended basket is defined as the purchased weekly basket, i.e. a vector of non-negative integer product quantities $\boldsymbol{x}^*_{k,q} \in \mathbb{N}_0^N$ for a specific household k at week q. In a real-world application, where the purchased basket is not known, the user may provide an intended basket via a shopping list interface or an e-shop basket interface. For now, week q and household k indices are omitted from the basket vector, as optimization calculations do not require values of intended baskets that were purchased in the past or from other households. The intended basket $\boldsymbol{x}^*$ is considered as the initial solution of the presented problem and is also considered as the basket that represents consumer taste and nutritional goals. For the current study we consider an ordered set C of $|C| = 11$ possible recommendation features, where a feature index $j = 1$ indicates the corresponding feature as shown in Table 4.2.

The synthetic dataset provides coefficients $c_{i,j}$, which in this study are median values over all transactions in the dataset and describe the corresponding feature j quantity $c_{i,j}$ per unit for each product i. Therefore, for a basket $\boldsymbol{x}$, one can calculate the total quantity for a specific feature as

$$v_j(\boldsymbol{x}) = \sum_{i=1}^{N} c_{i,j} x_i. \tag{4.1}$$

When designing the objective functions and comparing recommendations among baselines, we are interested in the ratio of total feature quantities between two baskets $\boldsymbol{x}, \boldsymbol{x}'$

$$\rho_j(\boldsymbol{x}, \boldsymbol{x}') = \frac{v_j(\boldsymbol{x})}{v_j(\boldsymbol{x}')} \tag{4.2}$$

j	Feature	Unit	Scope	Target
1	Cosine similarity	-	Personal	Max.
2	Cost	Dollars ($)	Personal	Min.
3	Energy	kilo Calories (kCal)	Personal	Pres.
4	Protein	grams(g)	Personal	Pres.
5	Fat	grams(g)	Personal	Pres.
6	GHG emissions	CO_2 kg eq.	Environment	Min.
7	Acidification pollution	SO_2 kg eq.	Environment	Min.
8	Eutrophication pollution	PO_4^{-3} kg eq.	Environment	Min.
9	Land use	m^2	Environment	Min.
10	Water usage	L	Environment	Min.
11	Stressed water usage	L	Environment	Min.

Table 4.2: Features relevant to objectives for basket selection. The first column shows the index of each feature, which also coincides with their position in the ordered set C. Feature, Unit, and Scope columns give a brief overview of the objectives and finally the Target column describes whether the goal of the optimization is to minimize, maximize, or preserve the intended basket value.

and more specifically the ratio of a recommendation towards the intended basket $\rho_j(\boldsymbol{x}, \boldsymbol{x}^*)$ for a specific feature j.

Individual Objectives

Consumer taste is the first personal value that is considered as an optimization objective, which is minimized when the recommended basket $\boldsymbol{x}$ is as similar as possible to the intended basket $\boldsymbol{x}^*$. High similarity between a recommended basket $\boldsymbol{x}$ and the target intended $\boldsymbol{x}^*$ indicates higher likelihood of a purchase under a counterfactual hypothesis, in which the user would consider recommended baskets before purchase. For this we define the first objective function to minimize, that is a function of the cosine similarity

$$J_1(\boldsymbol{x}, \boldsymbol{x}^*) = 1 - \frac{\boldsymbol{x}^\top \boldsymbol{x}^*}{\|\boldsymbol{x}\| \, \|\boldsymbol{x}^*\|} \tag{4.3}$$

Relation (4.3) is minimized by recommending the intended basket.

The next personal value considered in optimization is a function of cost. In general, it is assumed that individuals would prefer to minimize expenses and select cheaper baskets that satisfy their taste. Next, the cost ratio between recommended and intended basket costs is calculated as an objective function:

$$J_2(\boldsymbol{x}, \boldsymbol{x}^*) = \rho_2(\boldsymbol{x}, \boldsymbol{x}^*). \tag{4.4}$$

The intended basket does not minimize Relation (4.4), whereas a basket with no products at all would be the optimal solution.

Next, the nutritional values of a recommended basket are considered for optimization. For each unit of product i and nutritional product feature j the median nutritional quantity per unit $c_{i,j}$ is calculated. Three nutritional features are considered $j \in \{3, 4, 5\}$. The health objective functions use the intended basket nutritional value as a baseline to evaluate the difference for each nutritional feature between recommended and intended baskets:

$$J_j\left(\boldsymbol{x}, \hat{\boldsymbol{x}}\right) = \left(1 - \rho_j(\boldsymbol{x}, \boldsymbol{x}^*)\right)^2 = \left(\frac{v_j(\boldsymbol{x}^*) - v_j(\boldsymbol{x})}{v_j(\boldsymbol{x}^*)}\right)^2, j \in \{3, 4, 5\} \tag{4.5}$$

The intended basket is one solution that minimizes the Relation (4.5).

Environmental Impact Objectives

Collective environmental values are also considered based on the provided data from Ref. [11]. In total, a set of six environmental impact criteria are considered for each product, as shown in Table 4.2, namely *green house gas* (GHG) emissions, which contribute to climate change, *acidifying* pollution that decreases fertility and can cause desertification, *eutrophication* pollution, which destabilizes food chains in ecosystems, *water usage* that has several environmental effects, *stress-weighted water* usage that takes into account whether the water is taken from arid/dry lands, and *land usage*, which is important to resource allocation for farming and deforestation. The median product features per unit are used as coefficients $c_{i,j}$ for calculating $v_{\mathrm{j}}, j > 5$. Similar to the price objective, the ratio between intended and recommended basket of each environmental impact feature is considered as an objective function:

$$J_j\left(\boldsymbol{x}, \boldsymbol{x}^*\right) = \rho_j(\boldsymbol{x}, \boldsymbol{x}^*) \tag{4.6}$$

It is important to note that for the current dataset, we observe no negative values for any coefficient $c_{i,j}$, thus all nominators and denominators of the proposed objectives are positive. Unless the intended basket optimizes all of the above objectives simultaneously and is non-empty, then there is no solution that optimizes the above objectives simultaneously. For example this can be shown when removing a single item from an non-empty intended basket. The item removal will decrease the price objective value and also environmental impact objectives, while it will increase the taste objective value.

Problem Formulation

The proposed optimization framework evaluates the recommended baskets across a set $\mathrm{C} = \{1, 2, ...M\}$ of all *M=11* different objectives presented above. The optimization is performed in a decentralized manner and only uses the intended basket to decide objective function values. The multi-objective problem for M objectives can be summarized as

$$\min_{\boldsymbol{x}} \left(J_1(\boldsymbol{x}, \boldsymbol{x}^*), J_2(\boldsymbol{x}, \boldsymbol{x}^*), ...J_M(\boldsymbol{x}, \boldsymbol{x}^*)\right), \quad \boldsymbol{x} \in \hat{\mathrm{X}} \tag{4.7}$$

for a set of feasible baskets $\hat{\mathrm{X}}$. An optimization algorithm $f(\mathrm{X}_0; \boldsymbol{w}) = \mathrm{X}$ with parameter vector $\boldsymbol{w}$ takes an initial set of baskets X_0 and calculates a recommended set of baskets X. The goal of such algorithm is to find a non-dominated set of baskets. A basket $\boldsymbol{x}$ dominates $\boldsymbol{x} \prec \boldsymbol{x}'$ another basket $\boldsymbol{x}'$ if $J_j(\boldsymbol{x}) \leq J_j(\boldsymbol{x}')$ for all $j \in C$ and $J_j(\boldsymbol{x}) < J_j(\boldsymbol{x}')$ holds at least for one j [263]. If no other basket dominates $\boldsymbol{x}$, then it is referred as non-dominated.

4.2 Evolutionary Algorithms and Multi-Objective Optimization

Evolutionary algorithms are widely used for multi-objective optimization [264] with $M \geq 2$ objectives. A brief overview is illustrated in Figure 4.1a. Typically each basket, or solution[1] in the optimization context, $\boldsymbol{x}$ is mapped to an objective vector $\boldsymbol{\zeta}(\boldsymbol{x}) \in \mathbb{R}^M$, where each vector element represents an objective function value $\zeta_j = J_j(\boldsymbol{x})$. Often, such algorithms improve a set of an initial population of solutions X_τ by applying probabilistic operators on each solution vector $\boldsymbol{x}$, such as the *random crossover*. Random crossover randomly combines elements from different solutions $\boldsymbol{x}, \boldsymbol{x}'$ with probability p

$$x_i = \begin{cases} x_i' & \text{if } \delta < p \\ x_i & \text{otherwise} \end{cases}, \tag{4.8}$$

where δ is sampled from a probability density function $\delta \sim f$ with finite support $[0, 1]$. Another probabilistic mechanism is the *random mutation*, e.g. replacing an element of the solution with a random number sampled from a probability distribution $\kappa \sim f_{\text{discrete}}$ to an element of the solution

$$x_i = \kappa. \tag{4.9}$$

Each new solution is evaluated based on the corresponding objective vector $\boldsymbol{\zeta}(\boldsymbol{x})$ and a *selection* of solutions is performed. Typically a non-dominated sorting is performed for the selection of non-dominated solution candidates both from new solutions and the initial population. The non-dominated sorting is performed recursively, i.e. each time a non-dominated set F_α is selected, the non-dominated solutions are assigned to F_α and then removed from the population. A new non-dominated search is performed on the remaining solutions to determine the non-dominated front $F_{\alpha+1}$. This process repeats until all solutions are assigned to a front. A possible selection mechanism would select all non-dominated solutions, i.e. the solutions in F_1. The selected solution candidates are preserved in a new population of solutions $\mathrm{X}_{\tau+1}$ and the whole process (crossover, mutation, selection) is repeated until a convergence criterion is met, e.g. no new solutions are preserved in a population after an iteration. Often τ is referred to as a *generation*. A widely used genetic algorithm for multi-objective optimization is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [263].

[1] The term solution will be used in the sections that describe models in accordance to literature.

Gradient Guided Genetic Algorithm

Probabilistic algorithms are often criticized for slow convergence time [265], especially on high dimensional problems, and dependence on randomness [266, 267]. Recently, deterministic chaos genetic algorithms have been proposed to calculate solutions in a deterministic and seemingly in a more efficient manner [268, 269]. Furthermore, chaotic maps seem very promising for sparse and highly dimensional problems as they can control entropy [266] and the performance of the optimization procedures. For example, genetic algorithms may show improved performance if a logistic map [266] is used to sample initial solutions around the intended basket. Nevertheless, chaos genetic algorithms do not use explicit feedback from the loss function, such as Multi-Objective Natural Evolution Strategies [260] (MO-NES), and often the selection of adequate chaotic maps requires extensive hyper-parameter optimization [266, 267]. The current chapter investigates another potential design, where NNs are used to perform *mutation and crossover* operators and/or *initialize the population* instead of chaotic maps. NNs show promising capabilities to learn chaotic maps and strange attractors [270], and back-propagation can be used to learn the parameters of the NNs and control the chaotic behavior to improve solutions across generations.

Next, an overview of G3A is provided based on Figure 4.1b, but more technical details can be found in the Appendix Section C.4. An initial population matrix $\boldsymbol{X}_0$ is calculated by applying the untrained neural mutation from $t = 0$ to $t = T$. B solutions are selected during initialization, by sampling the mutation trajectory every $\Delta t = T/B$. During each generation, a population matrix $\boldsymbol{X}_\tau \in \mathbb{N}_0^{B\times N}$ is created, where each row represents a recommended solution.

A neural crossover operator is then applied on the population matrix and generates an offspring solution for each solution in the initial population. The main NN component is a transformer network $f_{\text{transformer}} : \mathbb{N}_0^{B\times N} \to \mathbb{R}^{B\times B\times N}$ with Gaussian Error Linear Unit [271] (GeLU) activation functions as hidden layers [272]. Each parent solution $\boldsymbol{x}$ is compared with the rest of the population matrix $\boldsymbol{X}_\tau$. For each element x_i of the parent solution, the transformer generates an attention vector $\boldsymbol{g} \in \mathbb{R}^B$ over all solutions in the population. The element $\hat{x}_{i,b}$ is selected from the b-th parent in the population that received the maximum attention value from the transformer $b = \arg\max_b g_b$. A sigmoid activation is then applied on the attention values and each selected parent element is used to calculate an "offspring" solution element x_i' in the following manner:

$$x_i' = \text{sigmoid}(g_k)x_i + (1 - \text{sigmoid}(g_b))\hat{x}_{i,b}. \tag{4.10}$$

Next, a mutation operator NN $u(\boldsymbol{x}(t)) : \mathbb{R}^{B\times N} \to \mathbb{R}^{B\times N}$ evolves a solution $\boldsymbol{x}(t)$ in continuous time t by applying the following neural ODE control

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{u}(\boldsymbol{x}(t)). \tag{4.11}$$

A neural ODE solve [136] scheme is used to calculate the continuous time evolution between subsequent genetic generations, e.g. $\boldsymbol{x}(0) \to \boldsymbol{x}(T)$. The underlying NN has sinusoidal activation functions in the hidden layer, inspired by the sinusoidal iterator used in Ref. [267]. To select the solutions that are preserved to the next

generation, a finite number of mutated solution is sampled uniformly across time for each solution $\boldsymbol{x}(0)$ of the current generation τ at predetermined time-steps within the ODE solver. The output activation of the NN is a Rectifier Linear Unit (ReLU) activation [272], which removes negative product quantities from each solutions.

NN weights and activation functions generate real-valued solutions. The proposed problems require discrete product quantities in the solution. Therefore, a discretization operation that allows gradient propagation is applied on each solution. A novel discretization scheme, termed fractional decoupling, is used and is described in detail in Section C.3. This scheme allows for gradient propagation during training, in comparison to rounding operators.

A non-dominated sorting is performed across all discretized solutions to determine the best solutions from each trajectory. The mean objective value per feature

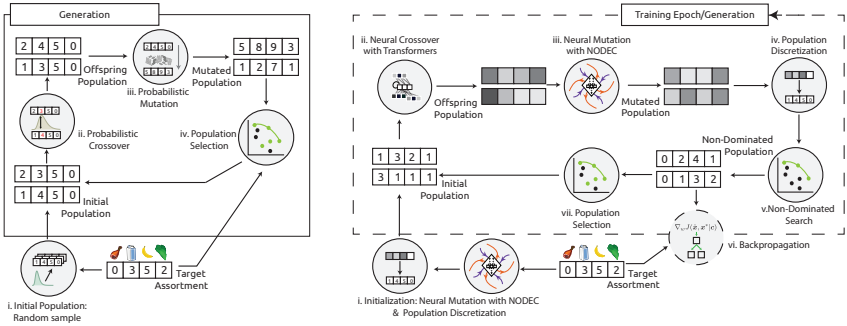
$$\overline{\zeta}_j = \frac{\sum_{\boldsymbol{x}(t)\in F_1} \zeta_j(\boldsymbol{x}(t))}{|F_1|} \tag{4.12}$$

is calculated over all non-dominated solutions, i.e. all samples $\boldsymbol{x} \in F_1$, and then each element $\overline{\zeta}_j$ is used to calculate gradients and perform the parameter update. Mean objective values $\overline{\zeta}_j$ can be scaled before gradient calculation to match user preferences and guide the algorithm towards non-dominated solutions that are better performing in specific objective values. To select the B solutions that are used as input population for the next generation, the hyper-volume and non-dominated ranks are used [260] as described in the MO-NES baseline in Appendix Section C.4. More technical details on training schemes, network architectures are found in the Appendix Section C.4.

4.3 Experimental Evaluation

Two multi-objective optimization algorithms are compared with G3A, namely MO-NES [260] and reference point NSGA-II [261] (RNSGA-II). Both baselines are described in more detail in Section C.4. All baselines are evaluated in weekly basket purchases that happen over the course of 85 weeks for 500 households, and in total 28400 intended baskets are considered. In particular, the households are chosen based on their total green house gasses (GHG) emissions, i.e. the top 500 emission producers are selected. G3A is parameterized to generate $B = 8$ recommendations per intended basket, whereas RNSGA-II and MO-NES generate $B = 10$ recommendations per intended basket. The population sizes where chosen after evaluating different values. The sizes that generated well-performing solutions efficiently were preferred. For each recommendation, a ratio towards the cost, environmental impact, or nutritional quantities of the intended basket are considered. Some of the ratio functions coincide with the proposed objective functions, but this is not the case for nutritional losses, as the normalized MSE showed better convergence, but required scaling.

It is important to note that all three baselines were tested on a subset of potential hyper-parameters. Hyper-parameter optimization was performed for several days to the extend that each method was able to solve the problem effectively. From



(a) GA

(b) G3A

FIGURE 4.1: A high level illustration of the G3A and the classic GA. In the classic GA (a), the components of initial population sampling (a.i), crossover (a.ii), and mutation (a.iii) are often performed in a probabilistic manner. The population selection (a.iv) applies various mechanisms, such as a non-dominated sorting, over the current population to select the solutions that will continue in the next generation. Often a subset of the non-dominated solutions is preserved to the next generation. In G3A (b.) the initial population sample (b.i), the crossover (b.ii), and mutation operators (b.iii) are performed with a NN. A discretization technique that allows back-propagation is then applied (component b.iv), and then a gradient descent update is performed based on the loss values of all non-dominated solution (components b.v-b.vi). Population selection (b.vii) happens after the network update, thus gradient guided genetics preserves information from all non-dominated solutions in its weight.

observed models, the best performing parameterization per method was selected. In future work, G3A will be compared against other optimization methods on more established problems to determine performance in terms of optimality. Such a study was out of the scope of this thesis.

Recommendation Comparison

First, the ability of baselines to produce non-dominated solutions for the problem is evaluated. Table 4.4 contains a comparison where all recommendations for an intended basket $\boldsymbol{x}^*$ from all methods are compared against each other and only the non-dominated solutions are kept across all methods. The ratio of total dominated solutions divided by total recommendations per method is calculated. All three baselines produce diverse non-dominated solutions, as they all achieve high mean ratio of non-dominated to total recommended baskets per intended basket. This indicates that the problem can be tackled effectively by all methods.

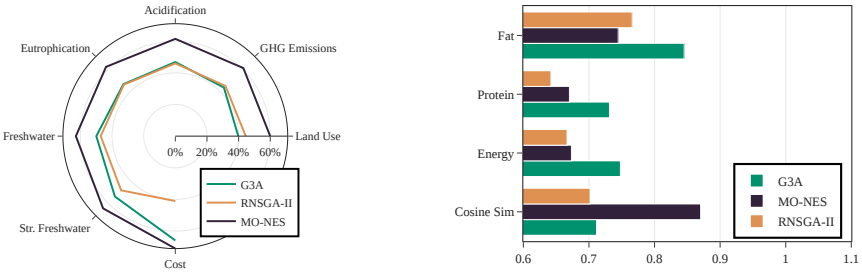
Model	Mean	Mean CI	Median	Median CI
G3A	0.980	(0.979, 0.981)	1.0	(1.0, 1.0)
MO-NES	0.948	(0.946, 0.949)	1.0	(1.0, 1.0)
RNSGA-II	0.986	(0.985, 0.986)	1.0	(1.0, 1.0)

Table 4.4: Mean and median values of non-dominated percentage of solutions when recommendations from all methods are combined together. Reverse bootstrap confidence intervals with significance level $\alpha = 0.05$ are also provided. All three baselines find a high percentage of non-dominated solutions, even when compared to each other.

Several recommended baskets per model may have non-preferred objective values. For example, a solution may achieve the optimal value in terms of a nutritional loss and then be selected as a non-dominated solution, although it produces 200% more emissions. Such solutions are discarded for the comparisons in the next sections, i.e. not recommended. A filtering is applied by discarding any solution that has a cost or any environmental impact quantity or cost ratio $\rho(\boldsymbol{x}, \boldsymbol{x}^*) \geq 1.0$. Furthermore, very dissimilar baskets are also discarded, i.e. when cosine_sim$(\boldsymbol{x}, \boldsymbol{x}^*) \leq 0.5$. For each recommendation the cosine similarity and environmental impact, nutritional and cost ratios towards the corresponding intended basket are calculated. The mean value of the ratio calculations over all recommendations per model are reported in Figure 4.2. Figure 4.2 indicates that RNSGA-II outperforms other baselines in terms of cost, while G3A shows higher performance in terms of nutritional values. MO-NES shows higher performance in terms of cosine similarity. As indicated by the dominance analysis results, all models can provide highly dominant solutions, that potentially specialize better in subsets of objectives. Depending on the design goals of the system or the priority of the individual, a different algorithm might be more preferred. Furthermore, all three models can be further altered to include consumer input in which objectives need to be prioritized. For G3A and MO-NES this can be implemented by adding weights and scaling the objective function values before the gradient update. For RNSGA-II this can be achieved by creating reference points that correspond to the user priorities.

Calculation Execution Time and Emissions

A sample of 100 intended baskets over a single week is used to determine execution time and calculation GHG emission for each model (see Table 4.6). Although G3A requires higher computation time and generates more emissions per calculation of recommendations, all models produce emissions and wall clock times are not significant. Accepting a single recommendation of any model can justify the emissions of thousands or even millions of calculations of other recommendations. Furthermore, G3A code is still at an experimental level, and better code optimization can be achieved to further reduce calculation times and emissions.



(a) Cost and Environmental Impact Ratios (lower values - points closer to center are preferred).

(b) Nutritional ratios and cosine similarity (higher values - longer bars are preferred).

FIGURE 4.2: A comparison of cosine similarity and the total emission, nutritional, and cost of a recommendation, as a ratio to the corresponding intended basket. For each baseline the mean ratio value over all recommendations that achieve cosine similarity higher than 0.5 and have all environmental ratios costs below 1.0 are considered.

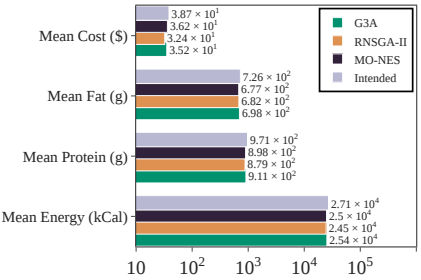
Model	Elapsed Wall Clock Time seconds	GHG Emissions kg CO_2 eq.	Mean, Min GHG Emission Improvement kg CO_2 eq.
G3A (GPU)	1.89 ± 1.22	2.07 ± 1.44)e-08	31.49, 0.46
MO-NES (CPU)	0.20 ± 0.01	(2.16± 0.14)e-09	21.03, 0.41
RNSGA-II (CPU)	0.46 ± 0.06	(6.95±2.41)e-10	34.04, 0.45

Table 4.6: Execution time and GHG emissions (mean ± standard deviation) measured with python and the codecarbon library [273] over a sample of 100 intended baskets from different households. The mean and minimum GHG emission improvement for accepting a single recommendation is also reported to outline the potential cost-benefit of accepting versus calculating recommendations.

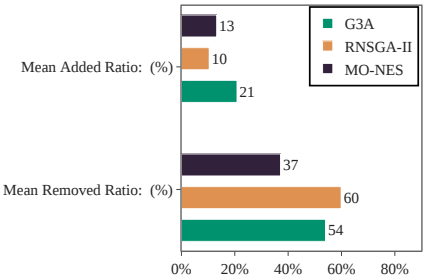
Real-World Impact

To extend the comparison of G3A and estimate the impact on total reduction values, a counterfactual scenario is evaluated. For each model, 5000 counterfactual trajectories are sampled, each trajectory being 86 weeks long. For each trajectory, it is assumed that 25% of all intended baskets are replaced with a recommendation. The recommendation which replaces the intended basket is chosen randomly[2]. Figure 4.3 illustrates the ability of all algorithms to achieve a considerable reduction of environmental impact compared to the intended basket. For example, deciding to replace 25% of intended baskets with a G3A recommendation leads to a reduction of approximately 35 metric kilo-tons of CO_2 eq. or approximately 1 billion litres of stressed freshwater for G3A. The current results indicate that G3A achieves similar performance to RNSGA-II, but by removing less and adding more products. MO-NES instead produces recommendations that have the least impact on the consumer basket.

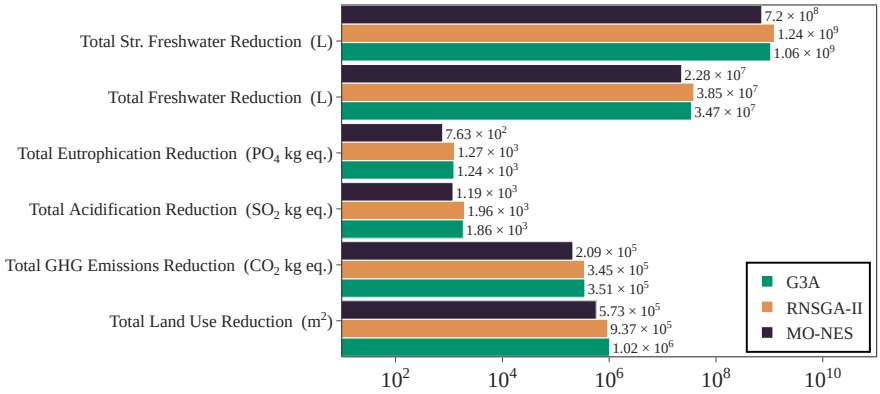
[2]In the current setting, a decision model for sampling, such as the one in [274] cannot be used, because the transactions of the current dataset may be effected by marketing campaigns and other covariates. Furthermore, it is not apparent of whether consumers were aware of sustainability issues when performing a purchase, thus the modeling of environmental impact decision factors may be invalid. Thus, designing a valid decision model to estimate the effect of a recommender system in this case is out of scope of this thesis and could potentially be considered as future work.



(a) Mean nutritional value and cost per basket per trajectory.



(b) Mean ratios of added and removed units per basket per trajectory.



(c) Mean reduction of total environmental impact of accepting recommendations versus purchasing only intended baskets per sampled trajectory. Longer bars perform better.

FIGURE 4.3: Comparison of the impact per model over 5000 trajectories, where 25% of the intended basket purchases is randomly replaced with a recommendation. Mean nutritional quantities per basket and trajectory are reported (see a.). Next (see b.), the mean value of added and removed units per basket are provided over all recommendations and trajectories, where intended baskets are omitted for the calculation. The total environmental impact and cost reduction are calculated per sample and then subtracted from the total quantities of the original trajectory (only intended baskets are purchased). Although confidence intervals are calculated, they are omitted as they are mostly too narrow and, thus, not visible.

4.4 A Hybrid Design Approach

In this section, a different system design approach to G3A is evaluated, termed as Hybrid-G3A. This approach adds another computation layer after recommendations are calculated and can be considered applicable to other baselines as well. A hybrid AI design method is preferred, in which both centralized and decentralized design paradigms are combined [51]. Not all intended baskets from all 500 households are now considered for recommendation in Hybrid-G3A. Instead, a subset of intended baskets is filtered out by comparing the average environmental impact features per person per intended basket, to calculate a global baseline for each week. The provided dataset contains data at household level, therefore in order to apply global constraints, a more preferable approach would compare environmental impact per person in the household[3]. We now identify an intended basket $\boldsymbol{x}^*_{k,q}$ and reuse the household index k and the week index q. Therefore, for an intended basket $\boldsymbol{x}^*_{k,q}$ purchased by a household k with ν_k persons, the mean environmental impact feature j per person is calculated as

$$\overline{v}_{j,k,q}(\boldsymbol{x}^*_{k,q}) = v_j(\boldsymbol{x}^*_{\boldsymbol{k},\boldsymbol{q}})/\nu_k. \tag{4.13}$$

For each $\overline{v}_{j,k,q}$ a masked value (following the method discussed in Chapter 5) is calculated by adding noise ϵ

$$m_{j,k,q} = \overline{v}_j(\boldsymbol{x}^*_{k,q}) + \epsilon \tag{4.14}$$

from a Laplace distribution $\epsilon \sim \mathcal{L}(\mu, \beta)$ with location parameter $\mu = 0$ and strictly positive scale parameter $\beta = |\overline{v}_j(\boldsymbol{x}_{k,q}| + 0.1$. The masked value is then send to a centralized system, which calculates the global mean environmental impact per person per week for K_q available households:

$$\overline{m}_{j,q} = \frac{\sum_{k=1}^{K_q} |m_{j,k,q}|}{K_q}. \tag{4.15}$$

Each intended basket $\boldsymbol{x}^*_{k,q}$ is compared against the global weekly average for each environmental impact feature, and a recommendation is proposed if

$$\frac{v_{j,k,q}(\boldsymbol{x}^*_{k,q})}{\nu_k} > 0.5\overline{m}_{j,q} \tag{4.16}$$

for any environmental impact feature $j > 5$. The coefficient 0.5 was chosen, so that at least 25% of all baskets can be selected for recommendations after all invalid baskets are rejected[4]. G3A considers only individual criteria to generate recommendations

[3]Seemingly not every household of interest has provided information regarding the number of persons in this household. These households are considered to have chosen a more privacy oriented attitude and are kept in resulting calculations. Still, Hybrid-G3A does not propose recommendations to them, assuming that non-disclosure of data is also an important value-sensitive constraint.

[4]Remember that similarity and environmental impact criteria also apply for recommended basket selection.

and it can be labelled as a decentralized system. The centralized system combines the privacy-preserving masks of the individual values and a centralized calculation. Then it disseminates the outcomes of the calculations to households, and only households with higher emissions per person are provided with recommendations for intended baskets. The "Hybrid-G3A" combines both systems, and therefore can be seen as a hybrid approach.

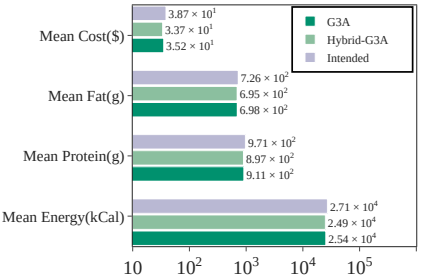
The analysis shown in Figure 4.3 is repeated for Hybrid-G3A and G3A and results are illustrated in Figure 4.4. A smaller sampling of 500 trajectories is performed, as confidence intervals do not seem to increase for larger samples. 25% of all intended baskets is uniformly sampled and replaced with a respective recommendation, either by considering (i) all intended baskets for replacements (G3A) or (ii) only the intended baskets that violate the global constraints of Relation (4.16) (Hybrid-G3A). The results in Figure 4.4 indicate that Hybrid-G3A surpasses G3A in terms of environmental impact (see Figure 4.4c), achieves similar performance in nutritional values (see Figure 4.4a), but also makes more product removals and additions to the intended baskets with higher emission values. On the contrary, approx. 68% of the all the intended baskets satisfy Relation (4.16) and the respective households do not need to consider recommendations and change their shopping patterns.

Possible future extensions of the hybrid approach would be a coupling of a more sophisticated centralized system to support recommendations with Hybrid-G3A. In such an approach, a centralized model such as NODEC (Chapter 2) could control a continuous time approximation of the Dynamic Integrated Climate Economy model [102] to determine per capita upper bounds for environmental impact variables in real time. Estimation errors on these bounds can be further corrected with privacy-preserving aggregates from consumers following Chapter 5. The upper bounds could be used to determine intended baskets that require Hybrid-G3A recommendations in a similar manner to the process discussed in this section.

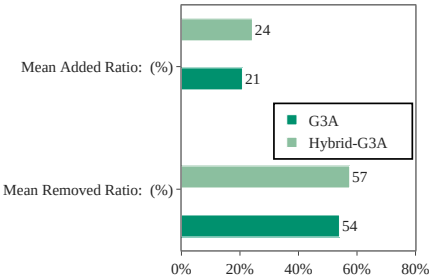
4.5 Conclusion

This chapter showcases a more quantitative approach to sustainable recommendations, where value-sensitive design is also taken into account. The problem of finding sustainable personalized baskets is revisited based on real-world data. Existing baselines are compared with a novel gradient guided genetic algorithm (G3A) and results showcase that all considered models produce good solutions to the problem. Even when individuals would adopt a fraction of the sustainable recommendations, a considerable environmental impact can be observed.

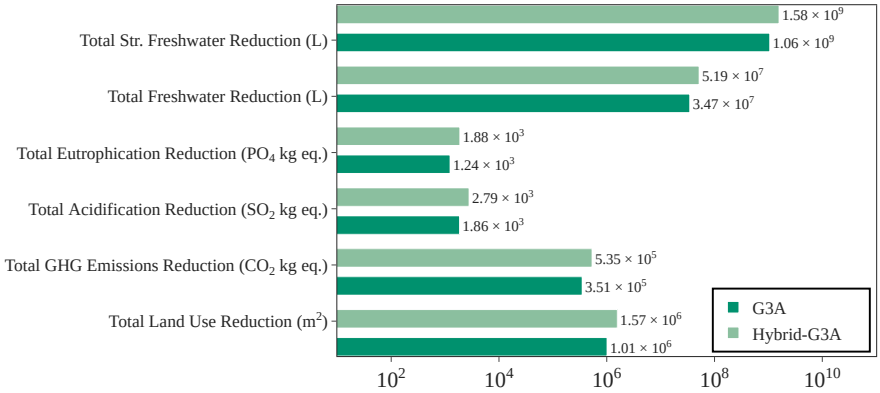
From a technical perspective this chapter introduces a novel multi-objective optimization algorithm that has comparable performance with state-of-the-art baselines on the new task. To make this happen, existing techniques from evolutionary methods are combined with state-of-the art NN architectures and a novel discretization technique, termed fractional decoupling. This technique allows for efficient gradient propagation when NNs are applied to mixed integer programming problems, such as the one presented in this chapter.



(a) Mean nutritional values per basket per trajectory.



(b) Mean ratios of added and removed units per basket per trajectory.



(c) Mean total environmental impact per sampled trajectory.

FIGURE 4.4: Comparison of the absolute impact per model over 500 purchase trajectories, where 25% of the intended basket purchases is randomly replaced with a recommendation, following the analysis presented in Figure 4.2. Although confidence intervals are calculated, they are omitted as they are mostly too narrow.

From a design perspective, the initial implementations G3A and other baselines follow a decentralized design, and then a hybrid implementation is provided. To effectively combine centralized and decentralized approaches, the privacy-preserving aggregation framework discussed in Chapter 5 is applied. The hybrid approach achieves higher performance in terms of environmental impact objectives but potentially introduces a higher amount of changes to intended baskets with higher environmental impact. Concluding, this chapter has showcased that implementation of existing or novel AI techniques can be performed with value-sensitive design.

Potential challenges can also be considered in future work, especially as convergence guarantees to desired optimal solutions are not always present when using genetic algorithms. Furthermore, the usage of a synthetic dataset may introduce errors, due to simplification and mismatches between real-world data-sets. Errors in the underlying data-sets can affect the optimization process and lead to undesired recommendations. Finally, the hybrid design approach may be further enhanced by including more sophisticated centralized mechanisms.

Part II

AI and Optimization Models

Chapter

5

Enabling Hybrid Design Approaches: Self-Determined Privacy-Preserving Collective Aggregations of Individual Data

Use of smart devices such as smartphones, wearables, and embedded sensors generates high data volumes in real time. Big data systems process these data, generate information, and enable services that support critical sectors of the economy, e.g. health, energy, and transportation systems etc. Such systems often rely on centralized servers or cloud computing systems. They are managed by corporate third parties referred to in this chapter as *data consumers* who collect the data of users referred to respectively as *data producers*. Data consumers perform data analytics for decision-making and automation of business processes. However, data producers are not always aware of how their data are used and processed. Terms of Use are shown to be limited and ineffective [275, 276]. Security and privacy of users' data depend entirely on data consumers. As a result misuse of personal information is possible, for instance, discrimination or limited freedom and autonomy by personalized persuasive systems [16, 17, 65, 66]. Giving control back to data producers by self-regulating the amount/quality of shared data can limit these threats [277]. Incentivizing the sharing of a higher amount/quality of data results in improved quality of service, i.e. higher prediction accuracy [278, 279, 280]. At the same time, data sharing empowers data producers with an economic value to claim.

Several applications do not require storage of the individual data generated by data producers. Instead, data consumers may only require aggregated data. For instance, Smart Grid utility companies compute the total daily power load or the average voltage stability to prevent possible network failures, bottlenecks, predict future power demand, optimize power production, and design pricing policies [281, 282]. Privacy-preserving masking mechanisms [283], i.e. differential privacy, accurately approximate the actual aggregate values without transmitting the privacy-sensitive individual data of data producers. Masking is a numerical transformation of the sensor values that usually relies on the generation of random noise and is irreversible[1].

The studies conducted in Chapters 2 and 3 had access to aggregates or model estimates of individual (or data producers) data. Such data in reality is not always open and accessible to the respective data consumers, who were mainly the system

[1]It is computationally infeasible to compute the original data using the transformed data.

designers or the shareholders that perform research and evaluation on the system. For example, in Section 2.4 the infection fraction of individuals is accessible to NODEC. Also, the preference profiles in Section 3.1 were extracted in non-privacy-preserving manner, while they provide useful information for sustainability experts or food producers. Access to accurate aggregates of individual data with privacy-preserving masking can contribute to enhance value-sensitive design of both centralized and decentralized approaches. Especially the combinations of both design approaches, where centralized systems receive data aggregates to perform institutional decisions and disseminate information to decentralized systems that support individual decisions as well, as shown in Section 4.4. This chapter aims to effectively bridge centralized and decentralized approaches based on a self-determined method by allowing individuals to choose for themselves which privacy masking setting suits their preferences better.

Privacy-preserving masking mechanisms are studied by calculating metrics of privacy q and utility u. The former represents the amount of personal information that a data producer preserves when sharing a masked data value. The latter represents the benefit that a data consumer preserves when using certain masked data for aggregation, e.g. accuracy in data analytics. Literature work [283, 284, 285] shows that privacy and utility are negatively correlated, meaning that an increase on one results in decrease on the other. This chapter studies the optimization of computational trade-offs between privacy and utility that can be used to model information sharing as supply-demand systems run by computational markets [277, 286]. These trade-offs can be measured by the opportunity cost between privacy-preservation and the performance of algorithms operating on masked data, i.e. prediction accuracy. Trade-offs can be made by choosing different parameters for different masking mechanisms each influencing the mean or the variance of the generated noise distributions [283]. Each parameterization results in a pair of privacy and utility values within a trajectory of possible privacy-utility values.

The selection of parameters for masking mechanisms that maximize privacy and utility is studied in this chapter as an optimization problem [284, 285]. In contrast to related work that exclusively focuses on universal optimal privacy settings (homogeneous data sharing), this chapter studies the optimization of privacy-utility trade-offs under diversity in data sharing (heterogeneous data sharing). This is a challenging but more realistic scenario for participatory data sharing systems that allow informational self-determination via freedom and autonomy in the amount/quality of data shared by each data producer. A novel computational framework is introduced to compute the privacy settings that realize different privacy-utility trade-offs.

The main contributions of this chapter are the following: (i) The introduction of a generalized, domain-independent, data-driven optimization framework, which selects privacy settings that maximize privacy and utility. (ii) A formal proof on how high utility can be achieved under informational self-determination (heterogeneous data sharing) originated from the diversity in the privacy settings selected by the users. (iii) The introduction of new privacy and utility metrics based on statistical properties of the generated noise. (iv) The introduction of a new masking mechanism.

(v) An empirical analysis of privacy-utility trajectories of more than 20,000 privacy settings computed using real-world data from a Smart Grid pilot project.

This chapter is outlined as follows: Section 5.1 includes related work on privacy masking mechanisms, privacy-utility trade-off as well as privacy-utility maximization problems. Section 5.2 defines the optimization problem and illustrates the research challenge that this chapter tackles. Section 5.3 introduces the proposed optimization framework. Section 5.4 outlines the experimental settings on which the proposed framework is tested and evaluated. Section 5.5 shows the results of the experimental evaluation. Finally, Section 5.6 concludes this chapter and outlines future work.

5.1 Related Work

In the past, several algorithms have been proposed to perform data aggregation without transmitting the raw data. The basic idea behind such algorithms is to irreversibly transform[2] the data, so that the original values cannot be estimated. While doing so, some of the properties of the data should be preserved to accurately estimate aggregation functions such as the count, sum or multiplication [283, 279, 277, 287, 288]. The masking process enables the data producers to control the amount of personal information sent to data consumers. These methods also ensure that the data remain private even when a non-authorized party acquires them, for example in the case of a "man-in-the-middle attack" [288].

Privacy-Preserving Mechanisms

An overview of privacy-preserving mechanisms is illustrated below:

Perturbative Masking Mechanisms

Perturbative masking mechanisms allow the data producers to share their data after masking individual values. Each value is perturbed by replacing it with a new value that is usually generated via a process of random noise generation or vector quantization techniques on current and past data values [283]. Some of the most well-known perturbative masking methods are the following:

Additive noise: A privacy-preserving approach is the addition of randomized noise [288, 94, 289]. This approach is often used in differential privacy schemes [94]. Differential privacy is ensured when the masking process prohibits the estimation of the real data values, even if the data consumer can utilize previously known data values or the identity of the individual who sends the data [290]. Algorithms that achieve differential privacy rely on the notion that the change of a single element in a database does not affect the probability distribution of the elements in the database [288, 290, 289, 291]. Furthermore, the removed element cannot be identified when comparing the version of the database before and after the removal. This is achieved by adding a randomly generated noise to each data value. The distribution

[2] A process also known as masking.

of the random noise is parameterized and usually is symmetric around 0 and relies on the cancellation of noises with opposite values. Increasing the number of noise values also increases the noise cancellation, since a larger number of opposite values are sampled. This property can be used to combine differential privacy mechanisms in order to ensure privacy while achieving high utility [292]. Statistical aggregation queries on the masked data return an approximate numerical result, which is close to the actual result. Differential privacy can be applied to discrete and continuous variables for the calculation of several aggregation functions [279]. Differential privacy can be combined with the usage of deep neural networks [293, 294], to apply more complex aggregation operations on statistical databases. Furthemore, several additive noise implementations are susceptible to noise filtering attacks, such as the use of Kalman filters [295] or reconstruction attacks [296]. These attacks can be prevented when the noise is not autocorrelated or the distribution of its autocorrelation is approximately uniform.

Microaggregation: Microaggregation relies on the replacement of each data value with a representative data value that is derived from the statistical properties of the dataset it belongs to. A well-known application of microaggregation is K-anonimity. K-anonymity relies on the notion that at least K original data values are mapped to the same value [297]. When a crisp clustering algorithm is applied on the data, each data value is mapped to the cluster centroid it belongs to. K is the minimum number of elements in a cluster. Using crisp clustering techniques[3] may result in vulnerabilities to specific attacks, so membership or fuzzy clustering is preferred instead [298]. Membership clustering assigns a data point to multiple clusters with a probability that is often proportional to the distance from each cluster centroid. For membership clustering techniques, usually large amounts of data are required. The storage and computational capacity of sensor devices cannot usually support such processes [298, 283].

Synthetic microdata generation An new dataset is synthesized based on the original data and multiple imputations [283]. The "synthetic" dataset is used instead of the original one for aggregation calculations. The application of synthetic microdata generation on sensor devices may produce prohibitive processing and storage costs. Furthermore, the availability of historical data on each sensor device may not be adequate for such methods to achieve comparable performance and efficiency with the perturbative masking methods [283].

Encryption

Several approaches use encryption to produce an encrypted set of numbers or symbols, known as ciphers. The aggregation operations can be performed on the ciphers and produce an encrypted aggregation value. The encrypted aggregate value can then be decrypted to the original aggregate one, with the usage of the corresponding private and public keys and decryption schemes, providing maximum utility and privacy to the recipient. The encrypted individual values cannot be transformed to the original values without the usage of the appropriate keys from an adversary, so

[3]Such as K-Means.

maximum privacy is ensured. Currently, there is extensive research on this area, and there has been a recent breakthrough with the development of fully homomorphic encryption schemes [299, 300, 287, 301]. Homomorphic encryption schemes though require high computational and communication costs, especially when applied in large scale networks [300, 302].

Multi-Party Computation

Multi-Party Computation (MPC) [303, 304] can also be used for privacy-preservation [305] by moving data from one device to another. In such an approach, security and integrity of the data depend on the resilience and security of the network. Most of the methods that rely on encryption can calculate the exact sum of the data, but they can also be violated if an attacker manages to have access to the private key or uses an algorithm that can guess it. Furthermore, in most cases they rely on communication protocols that burden the system with extra computational and communication costs [306]. These costs are often prohibitive for devices such as IoT sensors and smartphone wearables in which computational power and storage are limited [304].

Privacy and Computational Markets

A supply-demand system operating on a computational data market, can be created with the introduction of self-regulatory privacy-preserving information systems [277]. Privacy preservation is utilized to create such systems, for instance by using K-means for microaggregation and different numbers of clusters for each sensor. Varying the number of clusters produces different levels of privacy and utility. The resulting trade-off between privacy and utility is used to create a reward system, where data consumers offer rewards for the data provided by the data producers. The rewards are based on the demand of transformed data that enables the estimation of more accurate aggregate values.

A reward system can be combined with pricing strategies from existing literature on pricing private data [286], in which three actors are introduced: Various pricing functions are proposed to the *Market Maker* so that the privacy-utility of both data consumers and data producers are satisfied. The optimization framework of the current chapter can utilize any parametric masking mechanism of the literature mentioned in Section 5.1. The output of the optimization can be used along with pricing functions on participatory computational markets, to create fully functional and self-regulatory data markets.

Comparison and Positioning

The challenge of an automated selection of privacy settings that satisfy different trade-offs is not tackled in the aforementioned mechanisms. Privacy-utility trajectories have not been earlier studied extensively and empirically as in the rest of this chapter. The optimization of privacy-utility trade-offs under diversity in data sharing originated

from informational self-determination is the challenge tackled in this chapter. To the best of the authors' knowledge, this challenge has not been the focus of earlier work.

5.2 Problem Definition

Related work [285, 278, 284, 307, 277] on privacy-utility trade-offs focuses on the parameter optimization of a single masking mechanism. A masking mechanism is often a noise generation process, which samples random noise values from a Laplace distribution and then it aggregates it to the data, for instance the sampled noise is then added to the data to achieve differential privacy [288]. The result of the optimization is usually a vector of parameter values $\theta_{\eta,k}$, for a masking mechanism η and parameter index k. The pair of the masking mechanism and the parameter values is referred as a *privacy setting* $f_\eta\left(S, \theta_{\eta,k}\right)$ of a set of sensor values $S \in R^1$. This privacy setting produces a pair of privacy-utility values q^*, u^*, such that:

$$q^* \rightarrow max\left(Q\right) \tag{5.1}$$

$$u^* \rightarrow max\left(U\right) \tag{5.2}$$

Where (q^*, u^*) is a (sub-optimal) privacy-utility pair of values, which is computed by an optimization algorithm that searches for the optimal privacy-utility values pair. $max\left(Q\right)$, $max\left(U\right)$ are the maximum privacy and utility values of a privacy value set Q and a utility value set U. These sets are generated by the application of a masking mechanism.

The optimization of an objective function that satisfies both Relations (5.1) and (5.2) simultaneously is an NP-hard problem [285], in the case that privacy and utility are orthogonal ($q \perp u$) or opposite[4] ($q \uparrow, u \downarrow$), and often intractable to solve, since privacy-utility trade-offs prohibit the satisfaction of both Relations (5.1) and (5.2). Particularly, maximizing simultaneously utility and privacy usually yields sub-optimal values, which are lower than the corresponding optimal values computed by optimizing each metric separately [285]. Furthermore, such optimization is applicable for statistical databases [290, 283], where data are stored in a centralized system. In such case, a specific privacy setting is chosen by the designer/administrator of the system. As a result, this approach relies on the assumption that a specific privacy setting should be used by all data producers.

However, remaining to a fixed privacy setting may be limited for data producers, especially when a data producer wishes to switch to a different privacy setting to improve privacy further. In this case, the optimization of different objective functions is formalized in the following inequalities:

$$q^* > q^* + \delta \wedge u^* > u^* + c \tag{5.3}$$

[4]In the case that privacy and utility are positive correlated ($q \uparrow, u \uparrow$), the problem is reduced to NTIME-hard, and especially in the case privacy and utility are proportional $q \propto u$ to DTIME-hard [308]. The solution of the problem is provided by linearly evaluating all pairs of privacy and utility values once without comparing to all other pairs.

Where δ measures the change in privacy, which denotes whether the data producers require higher privacy, $\delta > 0$, or lower privacy $\delta < 0$, from the system. c measures the change in utility, which denotes whether the data consumer demands lower utility, $c > 0$, or higher utility $c < 0$, from the system. Finally, (q^*, u^*) denotes a new (sub-optimal) pair of privacy-utility values, computed by an optimization algorithm that searches for the optimal pair of privacy-utility values with respect to the privacy requirements of data producer and the utility requirements of data consumer expressed by c and δ respectively.

The optimization of an objective function to satisfy Relation (5.3) is also based on the assumption that all data producers agree to use the same privacy setting. This means that data producers may acquire a different privacy level by changing the value of δ via the collective selection of a different privacy setting. Consequently, a single privacy setting is generated and it produces a pair of privacy-utility values, which satisfy Inequality (5.3). The value of δ is determined via a collective decision-making process applied by the data producers, e.g. voting between different privacy-utility requirements. Such a system is referred to as a *homogeneous* privacy system, where data producers are able to influence the amount of privacy applied on the data by actively participating in the market, nevertheless they all share the same value for δ. The data consumer can bargain for higher utility by offering higher financial incentives to the data producers to lower their privacy requirements.

Another challenge that arises is the optimization between privacy and utility when each user decides and self-determines a preferred privacy setting instead of using a universal privacy setting. In such a scenario, inequality (5.3) is substituted by the following set of inequalities:

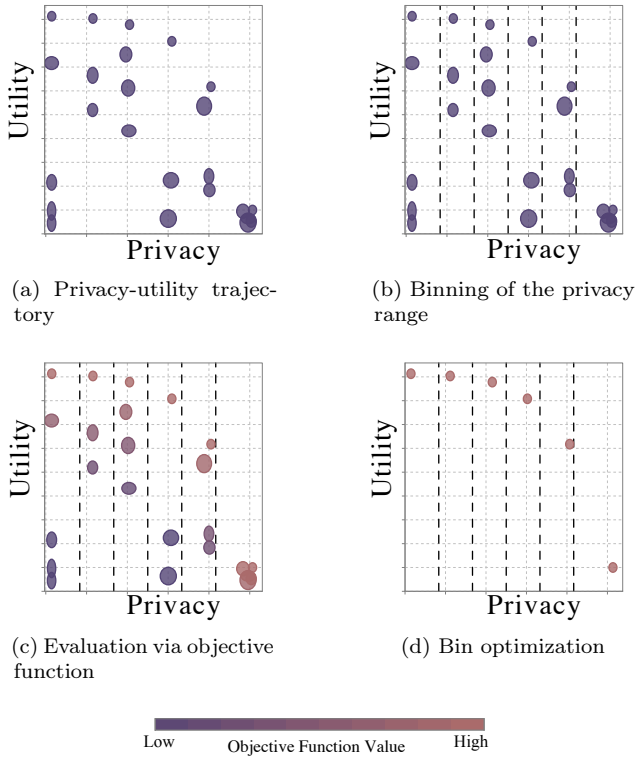
$$\begin{aligned} (q_1^* > q^* + \delta_1) \wedge \ldots \wedge \\ (q_n^* > q^* + \delta_{|N|}) \wedge (u^* > u^* + c) \end{aligned} \tag{5.4}$$

Where δ_n measures the change in privacy which denotes whether a data producer n belonging to a set of users N requires higher privacy, $\delta_n > 0$, or lower privacy $\delta_n < 0$. q_n^* denotes a new (sub-optimal) privacy value for each data producer n. The value is computed by an optimization algorithm that searches for the optimal privacy value with respect the data producer's privacy requirements expressed by δ_n.

A system in which the inequalities of (Relation (5.4)) hold is referred to as an *heterogeneous* privacy system, where each data producer self-determines and autonomously applies a privacy setting based on a preferred privacy value and an expected reward for increasing system utility.

5.3 Framework

The design of a new privacy-preserving optimization framework is introduced in this section to tackle the challenges posed in Section 5.2. Additive noise masking mechanisms require a lower number of parameters in general and they are often used in privacy-utility optimization [283, 290, 285]. Each privacy setting is illustrated as



(a) Privacy-utility trajectory

(b) Binning of the privacy range

(c) Evaluation via objective function

(d) Bin optimization

(e) Objective function scale

FIGURE 5.1: A graphical representation of the algorithm. Each ellipse denotes the privacy-utility values of a privacy setting. In Figures 5.1c and 5.1d the varying color denotes the fitness value. A lighter red color denotes higher fitness.

an ellipse[5] in Figure 5.1a. Each point within the ellipse is a possible privacy-utility pair of values. The ellipse center is chosen based on the privacy and utility mode of the setting. The mode is the value with the highest density. In symmetric distributions, it can be measured via the mean. The vertical radius of the ellipse denotes the dispersion of utility values, while horizontal radius denotes the dispersion of privacy values. Additive noise is stochastic, which means that applying the same privacy setting on the same dataset yields varying privacy-utility values. The choice of an optimal privacy-utility pair cannot be achieved by only evaluating the mode of privacy and utility for each privacy setting. If the privacy-utility values of a privacy

[5]The elliptical shape is chosen for the sake of illustration and it indicates a symmetrical distribution of privacy-utility values, generated by a privacy setting, within the ellipse area.

setting with high utility mode are varying to a large extend, there is high probability that unexpected non-optimal values are observed. To overcome this challenge, the objective function of the parameter optimization algorithm selects the parameters that minimize the dispersion[6] of privacy-utility values while maximizing the expected utility.

A data producer selects any privacy setting, among different ones, that satisfies personal privacy requirements. The proposed framework divides the range of privacy values in a number of equally sized bins, as illustrated in Figure 5.1b. Within each bin, a fitness value is calculated for each privacy setting, based on privacy-utility mode and dispersion. Each privacy setting produces privacy values with low dispersion. This is done by applying a lower bound constraint on privacy and utility constraint on the dispersion of privacy values and evaluating only privacy settings that satisfy this constraint, as shown in Figure 5.1c. The optimization framework evaluates several privacy settings, to find the parameters that achieve maximum privacy-utility values that vary as little as possible. This is illustrated in Figure 5.1d in which the ellipses with the highest utility mode and lowest utility dispersion are filtered for each privacy bin.

In a homogeneous data sharing system, a universal privacy setting is selected by the data producers, via, for instance, voting [309]. Alternatively, in a heterogeneous system, the data producers self-determine the privacy setting independently. Theorem 1 below proves that aggregation functions can be accurately approximated (utility can be maximized) even if different privacy settings from the same of different masking mechanisms are selected.

Theorem 1. *Let the transformation of $|I|$ disjoint subsets of sensor values S_i into the respective subsets of masked values M_i using a certain privacy settings f_i for each such transformation. It holds that the aggregation of the generated multisets of masked values M_i approximates the aggregation of the sensor values multiset S_i:*

$$g(\bigcup_{i=1}^{|I|} M_i) \rightarrow g(S), \tag{5.5}$$

given that the commutative and associative properties hold between each of the privacy settings f_i and the aggregation function g.

Proof. Let a multiset of real sensor values $S \subseteq R^1$ and $|I|$ disjoint subsets of S such that:

$$\bigcup_{i=1}^{|I|} S_i = S, S_i \neq \emptyset \; \forall \, i \in \{1, ..., |I|\} \tag{5.6}$$

Let a privacy setting $f : S, \Psi \rightarrow M$ be a pairwise element operation between a set of sensor values S and a set of noise values Ψ, that transforms each sensor value

[6]This refers to the dispersion measures of the privacy and utility distributions. If the values belong to a gaussian distribution, then the standard deviation is used to measure the dispersion. Since this is not always the case, other measures of scale can be used, such as the Inter-Quantile Range(IQR).

$s \in S$ by aggregating it with a randomly selected noise value ψ from Ψ to produce a masked value m:

$$\begin{aligned} f(S,\Psi) = g(S \cup \Psi) = M \Leftrightarrow \\ f(s,\psi) = g(\{s,\psi\}) = m \end{aligned} \tag{5.7}$$

Let $g : A \rightarrow R^1$ be an aggregation function which aggregates all elements of real values multi-sets S, Ψ, $M \subseteq A \subseteq R^1$ into a single real value $g(A) = z_A \in R^1$. Assume that $g : A \rightarrow R^1$ is defined in a recursive manner so that it satisfies the following equation for a multiset A and any union of all possible combinations of disjoint subsets A_i that satisfy Relation (5.6):

$$g(A) = g(\bigcup_{i=1}^{|I|} A_i) = g(\bigcup_{i=1}^{|I|} g(A_i)) \tag{5.8}$$

According to literature [310] the family of aggregation functions that Relation (5.8) applies to is referred to as extended aggregation functions[7]. The pairwise operation between s and ψ in f is designed in such way that it satisfies the commutative and associative properties when combined with the pairwise operation of g:

$$g(f(S,\Psi)) \overset{5.7,5.8}{=} f(g(S), g(\Psi)) \tag{5.9}$$

where $g(\Psi) \rightarrow \iota$, ι is the strong neutral element of the extended aggregation function g, such that:

$$\begin{aligned} g(g(A) \cup \iota) \; = \; g(A) \; \Rightarrow \\ g(g(A) \cup g(\Psi)) \; \rightarrow \; g(A) \end{aligned} \tag{5.10}$$

This property is used in the noise cancellation of Section 5.1. Let $|I|$ multisets Ψ_i of noise that satisfy Relation (5.6), then the following relation holds:

$$\begin{aligned} g(M_i) = g(f(S_i,\Psi_i)) \overset{(5.9)}{\Leftrightarrow} \\ g(M_i) = f(g(S_i), g(\Psi_i)) \overset{(5.10,5.7)}{\Leftrightarrow} \\ g(M_i) \rightarrow g(S_i), \end{aligned} \tag{5.11}$$

which means that each noise multiset Ψ_i is generated in such a way that the aggregation of $g(M_i)$ approximates the aggregation of $g(S_i)$. An illustrative example is the Laplace noise used in the literature for the aggregation functions of count or summation [288, 302], which satisfies Relations 5.7, 5.8 and 5.10. Now it can be

[7] A subset of those functions are the averaging functions, which include aggregations such as the mean, weighted mean, Gini mean, Bonferoni mean, Choquet integrals etc.

proven that:

$$g(\bigcup_{i=1}^{|I|} f_i(S_i, \Psi_i)) \overset{Relation\ (5.8)}{=} g(\bigcup_{i=1}^{|I|} g(f_i(S_i, \Psi_i))) \overset{Relation\ (5.11)}{\iff} g(\bigcup_{i=1}^{|I|} M_i) \to g(\bigcup_{i=1}^{|I|} g(S_i))) \overset{Relation\ (5.6), Relation\ (5.8)}{\iff} g(\bigcup_{i=1}^{|I|} M_i) \to g(S) \tag{5.12}$$

Thus, Theorem 1 is proven. □

The practical implication of Theorem 1 is that the aggregation of sensor values is approximated by the aggregation of masked values produced by different privacy settings. The approximation stands as long as the noise values produced by the different privacy settings satisfy Relations 5.9 and 5.10. According to Relation 5.6, each subset of sensor values should be masked by one privacy setting. Regarding the complexity of these operations, applying the masking on top of sensor values is linearly depended to the number of sensor values $|S_i|$ assigned to each privacy setting. Due to Relation (5.6), applying the proposed framework in real time increases computational complexity by $O(|S|)$. The original values are not stored or transmitted at runtime, thus the storage and communication complexity does not change. During optimization all the privacy settings $i \in I$ are applied to a training set of sensor values S. In that case real sensor values are stored and transmitted as well along with the masked values for each setting. The storage and communication costs increase by $O(|I| \cdot |S|)$. The computation costs also increase to $O(|I| \cdot |S|)$, which is a quadratic complexity in the worst case $|I| = |S|$. In most real-world applications, it is safe to assume that the sensor values have considerably higher volume to the evaluated privacy settings $|I| << |S|$, thus the expected computational, storage and communication complexity are linear to the number of sensor values.

The framework can be applied as a multi-agent system. It requires two types of agents representing the data consumers and data producers. This scheme can be applied in both centralized and decentralized aggregation services, such as MySQL or DIAS [311]. Finally in both heterogeneous and homogeneous systems, the data consumer can influence the data producer's choice by offering a higher amount of reward to achieve a higher utility.

5.4 Experimental Settings

This section illustrates the experimental settings, which are used to empirically evaluate the proposed framework. A set of sensor values S is used for the evaluation. Each sensor value $s_{n,t}$ belongs to a user n and is generated at time t. For each sensor value, a privacy setting that operates on the device of the data producer masks the sensor value $f_\eta(s_{n,t}, \theta_{\eta,k})$ by using the masking mechanism η with parameters $\theta_{\eta,k}$. Two metrics are used to evaluate privacy and utility.

Privacy evaluation

The main metric, which is used to calculate privacy, is the difference of the masked value and the original value, which is defined as the local error:

$$\varepsilon_{n,t} = \left| \frac{f_\eta\left(s_{n,t}, \theta_{\eta,k}\right) - s_{n,t}}{s_{n,t}} \right| \tag{5.13}$$

For a privacy setting to achieve a high privacy, a data consumer should not be able to estimate the local error for the sensor values sent by data producers. This is achieved by choosing privacy settings that generate noise that is difficult to estimate. As it is shown in the literature [277, 285, 290, 283], the noise is difficult to estimate, if it is highly random and causes a significant change in the original value. To avoid noise fitering attacks, noise with low or no autocorrelation is generated. The range of autocorrelation values can be determined analytically when the noise generation function is defined. In case this is not possible, a metric quantifying the color of noise can be included in the objective function. Randomness is evaluated by measuring the Shannon entropy [312] $H\left(\mathcal{E}\right)$ of the local error for all local error values $\mathcal{E}$. The entropy is calculated by creating a histogram of the error values and then applying the discrete Shannon entropy calculation. Each bin of the histogram has a size of 0.001. The significance of change is measured by calculating the mean local error $\overline{\mathcal{E}}$ and standard deviation $\sigma_{\mathcal{E}}$. When comparing privacy settings, higher mean, variance and entropy indicate higher privacy [283]. In this chapter, the objective function that measures privacy for a privacy setting $f_{\eta,k}$ is defined as follows[8]:

$$\begin{aligned} q = \alpha_1 \frac{\overline{\mathcal{E}_{\eta,k}}}{max\left(\overline{\mathcal{E}_{\eta,k}}\right)} + \alpha_2 \frac{\sigma_{\mathcal{E}_{\eta,k}}}{max\left(\sigma_{\mathcal{E}_{\eta,k}}\right)} \\ + \alpha_3 \frac{H\left(\mathcal{E}_{\eta,k}\right)}{max\left(H\left(\mathcal{E}_{\eta,k}\right)\right)} \end{aligned} \tag{5.14}$$

Where $\alpha_1, \alpha_2, \alpha_3$ are weighting parameters used to control the effect of each metric in the privacy objective function. $max\left(\bullet\right)$ is the maximum observed value for a metric during the experiments. This value is produced by evaluating all privacy settings $f_{\eta,k}$. Dividing by this value, normalizes the metrics in $[0, 1]$, so that the objective function is not affected by the scale of the metric.

Utility evaluation

The utility of the system is estimated by measuring the error the system accumulates within a time period by computing an aggregation function $g\left(\bullet\right)$ on the masked sensor values. Examples of such aggregation functions are the daily total, daily

[8]The error function described in Relation (5.13) and Relation (5.15) is also known in literature as absolute percentage error (APE) [313]. The error values are easy to interpret, as APE measures the relative change of the sensor values and aggregate values by using masking. Yet, when the denominator of the function is approaching zero, then the absolute relative error cannot be calculated. If the sensor values are sparse, then another error function can be used, such as MAPE.

average and weekly variance of the sensor values. The accumulated error is referred to as global error and is defined as:

$$\epsilon_t = \left| \frac{g\left(M_t\right) - S_t}{g\left(S_t\right)} \right| \tag{5.15}$$

A sample set of global error values ϵ is created by applying the masking process for a number of time periods of the dataset. The mean, entropy and variance of the global error of a privacy setting $f_{\eta,k}$ is calculated over this sample. The mean global error $\overline{\epsilon_{\eta,k}}$ indicates the expected error between the masked and actual aggregate. The standard deviation $\sigma_{\epsilon_{\eta,k}}$ and the entropy $H\left(\epsilon_{\eta,k}\right)$ of the global error, indicate how much and how often the masked aggregate diverges from the expected value. Minimizing all three quantities to 0, ensures that the masked aggregate approximates the actual aggregate efficiently. Thus, after the global error sample is created for each privacy setting, the corresponding utility objective function is calculated:

$$\begin{aligned} u = 1 - \Bigg(& \gamma_1 \frac{\overline{\epsilon_{\eta,k}}}{max\left(\overline{\epsilon_{\eta,k}}\right)} + \gamma_2 \frac{\sigma_{\epsilon_{\eta,k}}}{max\left(\sigma_{\epsilon_{\eta,k}}\right)} \\ & + \gamma_3 \frac{H\left(\epsilon_{\eta,k}\right)}{H\left(max\left(\epsilon_{\eta,k}\right)\right)} \Bigg) \end{aligned} \tag{5.16}$$

Where the weighting parameters $\gamma_1, \gamma_2, \gamma_3$ are used to control the effect of each metric in the utility objective function. $max\left(\bullet\right)$ is the maximum observed value for a metric during the experiments. This value is produced by evaluating all privacy settings $f_{\eta,k}$. Dividing by this value, normalizes the metrics in $[0, 1]$, so that the objective function is not affected by the scale of the metric.

Recall from Section 5.3 that utility and privacy vary, when repeating the masking process for the same privacy setting and dataset due to the randomness of the noise. A large sample to measure this variance is created by applying each privacy setting over three times on the same dataset. Then the framework of Section 5.3 filters the privacy settings based on the mode and the scale of the privacy-utility sample, as illustrated in Figure 5.1c. The privacy-utility samples for a privacy setting may not follow a symmetrical or normal distribution[9]. As a result, the maximization of the following objective function is based on utility:

$$\text{perc}\left(U, 50\right) + \text{perc}\left(U, 10\right) \tag{5.17}$$

Where $perc(U, i)$ calculates the i^{th} percentile of a set of utility values U produced by the application of a privacy setting.

The factors that maximize Relation (5.17) are: (i) the value of the mode, which is assumed to be approximated by the median and (ii) the dispersion towards values lower than the median, which is expressed by adding the 10^{th} percentile to the median. The objective function evaluates the median and the negative dispersion

[9]It is confirmed in some experimental settings that some privacy settings generate samples of privacy-utility values that do not pass a Kolmogorov Smirnoff normality test [314], and are also non-symmetrical.

(10^{th} percentile) of utility values. Positive dispersion is not taken into account in the optimization, since the abstract objective of the optimization is to ensure the least expected utility of a privacy setting for the data consumers. The privacy is constrained by evaluating only privacy settings in which the 10^{th} percentile differs from the privacy median for at most ω, as shown in Inequality (5.18). The value of ω is constrained to be lower or equal to the bin size of the optimization to ensure low privacy dispersion:

$$\text{perc}\,(Q, 50) - \text{perc}\,(Q, 10) < \omega, \tag{5.18}$$

Where $perc(Q, i)$ calculates the i^{th} percentile of a set a set of privacy values Q produced by the application of a privacy setting.

5.5 Experimental Evaluation

The proposed framework is evaluated experimentally by applying it to a real-world dataset. Privacy and utility are evaluated using over $20,000$ privacy settings for empirical evaluation.

Electricity Customer Behavior Trial dataset

The "Electricity Customer Behavior Trial" ("ECBT") dataset [95] contains sensor data that measure the energy consumption for $6,435$ energy data producers. The data are sampled every 30 minutes daily for 536 days. For the proposed framework, a set of sensor values S of $|N| = 6,435$ users and $|T| = 536$ time periods. The total number of sensor values in the set is $|S| = 165,559,680$. The sensor data are considered private and the utility company managing the energy network uses them to calculate daily total consumption in the grid, to predict possible failures and plan power production. The daily total consumption is an aggregation that can be defined as the sum of all the sensor values generated during the day: $g\,(S_t) = \sum_{n=1}^{6435} s_{n,t}$. Around 10% of the daily measurements are missing values, and are not included in the experiments. The significance of the missing values reduces as the aggregation interval increases. Therefore, a daily summation is chosen over more granular summation.

During the experiments, the local error of Relation (5.13) results in a non-finite[10] number only for a low number of maskings. Hence, these values are excluded from the experiments, so that the calculation of finite local error values is feasible. Concluding, the proposed framework operates on 90% of the ECBT dataset.

Privacy Mechanisms

Among several masking mechanisms [283], two ones are used for the evaluation of the framework. Each mechanism is parameterized using the grid search algorithm[11] [315].

[10] The original sensor value is zero, therefore the result of Relation (5.13) is infinite for non-zero noise or indefinite for zero-noise.

[11] Also known as parameter sweep.

The majority of masking mechanisms are parameterized with real numerical values. A grid search discretizes these values, and then evaluates exhaustively all possible combinations of parameter values.

Laplace Masking Mechanism

This mechanism is widely used in literature [290, 283, 279]. The noise in the experiments of this chapter is generated by sampling a Laplace distribution with zero mean. The scale parameter b of the distribution is selected to ensure maximum privacy. Part of privacy can be sacrificed to increase utility if the privacy requirements from the data producers are not high. In this masking mechanism, this is achieved by reducing the b. The scale parameter for each Laplace masking setting, is generated from value $b = 0.001$ and during the parameter sweep the value increases by 0.001 until it reaches $b = 10$.

Sine Polynomial Masking Mechanism

This mechanism is introduced in this chapter. The mechanism generates random noise that can be added to each sensor value. Assume a uniform random variable υ. The noise generated from the introduced masking mechanism is calculated as follows:

$$m = \sum_{\xi=0}^{|\Xi|} [\theta_\xi \sin(2\pi\upsilon)]^{2\xi+1} \tag{5.19}$$

The coefficients of the polynomial are denoted as θ_ξ, and ξ denotes the index of the coefficient. Both the length of the polynomial $|\Xi|$ and the individual coefficient values can be tuned to optimize the resulting privacy-utility values of the masking mechanism. The generated noise is symmetrically distributed around zero, because the odd power of the sine function produces both negative and positive noise with equal probability. The sine function and its odd powers are always symmetrical towards the horizontal axis, meaning that $\left|[\theta_\xi \sin(2\pi\upsilon)]^{2\xi+1}\right| = \left|[-\theta_\xi \sin(2\pi\upsilon)]^{2\xi+1}\right|$. Hence, the integral of each factor is zero $\int_0^{1.0} [\theta_\xi \sin(2\pi\upsilon)]^{2\xi+1} d\upsilon = 0$. Therefore the distribution of generated values is symmetrical around zero for $\upsilon \in [0, 1]$, which denotes that the global error mean is approximating zero. Increasing the length of the polynomial and the values of its coefficients, increases the magnitude of the local error, without affecting the global error, indicating that higher utility can be achieved without sacrificing privacy. These properties make polynomials of trigonometric functions, such as sine and cosine, eligible candidates for additive noise optimization. By increasing the polynomial length and tuning the coefficient values, a larger space of privacy settings is searched to maximize privacy and utility.

Each coefficient is assigned to a value in the space $[0.01, 1.8]$. The grid search in that space starts with a step of 0.03 until the value of 0.3, to evaluate settings that create low noise. Then the step changes to 0.3 until the value of 1.8, to evaluate privacy settings that generate higher values of noise. The sine polynomial masking settings are generated by creating all possible permutations of these values for 5

coefficients. This yields around 10,000 masking settings. Preliminary analysis on the autocorrelation and the spectrograms of the proposed sine polynomial noise does not show autocorrelation and recurring patterns over different spectrograms[12].

Error Analysis

Each privacy setting that results from parameterization of the mechanisms is evaluated by analyzing the local and the global error that they generate on varying subset sizes of the ECBT dataset. By sampling varying sizes of the dataset, the utility and privacy dispersion metrics are evaluated on a varying number of sensor values, calculating the effect of varying participation in the system. To create a random subset of the ECBT dataset, a subset of users N_{test} is chosen. In each repetition the users are chosen randomly. All users use a universal privacy setting. The initial size of the subset is 50 users, and then it increases by 50 users until $|N_{\text{test}}| = 500$ users. Then, the size of the subset increases by 500 users until $|N_{\text{test}}| = 6,435$. This process generates several local and global error values. The average, standard deviation and entropy of the local error and global error are calculated for all samples generated from the above process. The empirical cumulative distribution function[13] (CDF) is shown for each metric in Figure 5.2.

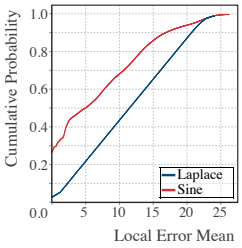
The sine polynomial mechanism can produce a wider range of local and global error values compared to the Laplace mechanism, since almost every sine polynomial CDF curve is covering a wider domain range on the domain axis compared to the respective Laplace CDF curves. The majority of the range axis values of the sine polynomial CDF curve are higher than the corresponding range values of the Laplace CDF curve. This indicates that it is more probable to generate lower global or local error value by using a sine polynomial setting compared to a Laplace setting. Concluding, the sine polynomial settings are expected to produce a wider range of privacy-utility trade-offs. Based on the CDF charts, sine polynomial settings are more likely to achieve higher utility, whereas Laplace settings are expected to achieve higher privacy.

Parameter Analysis

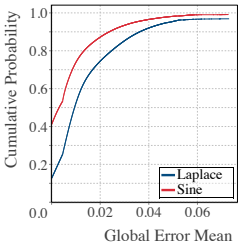
For the experiments, α and γ parameters are defined to calculate the privacy and utility. The choice of these parameters may vary based on the distribution of the sensor values and the kind of aggregation. Also data producers and data consumers may have varying requirements that affect the choice of those values. In this chapter, these values are determined empirically, to showcase an empirical evaluation. If a data consumer successfully calculates the mean value of the local error by acquiring the corresponding original values of a masked set, then it is possible to estimate

[12]Further analysis on this, is possible future work and is out of the scope of this chapter. This can be evaluated by introducing a metric that measures noise color in the privacy function.

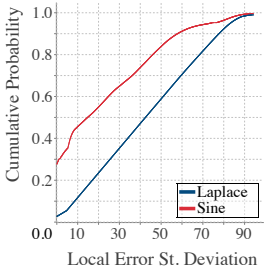
[13]The cumulative distribution function denotes the probability of a generated value being lower or equal than the corresponding domain axis value [316].



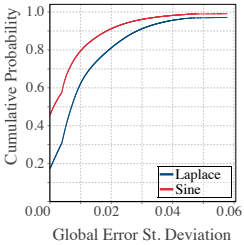
(a) Local error mean



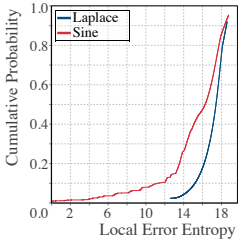
(b) Global error mean



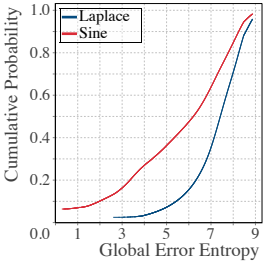
(c) Standard deviation of local error



(d) Standard deviation of global error



(e) Local error entropy



(f) Global error entropy

FIGURE 5.2: Cumulative distribution function of each local and global error metric computed for all settings of each masking mechanism.

the original sensor values of other masked sets as well by subtracting the calculated mean. This challenge is addressed by using privacy settings with high noise variance. Still, high variance does not guarantee that the masking process is not irreversible. If noise varies between a small finite number of real values, then the data consumer can also estimate the original value of the data by subtracting the variance. To overcome this challenge, privacy settings that produce noise with high entropy, therefore high randomness, are chosen. Consequently, a lower value for the coefficient of local error

mean is chosen as $\alpha_1 = 0.2$, while entropy and standard deviation of the local error share a higher coefficient value of $\alpha_2 = \alpha_3 = 0.4$.

Assigning values to the utility coefficients depends highly on the preferences of the data consumer. In the case of sum, the global error mean should be near 0, unless the data consumer estimates the mean and then subtracts it from the aggregation result. For this chapter the main concern is to keep a global error mean near zero, to avoid the aforementioned correction process. Standard deviation and entropy are assigned with equal weight. Therefore, a very high coefficient of $\gamma_1 = 0.6$ for the global error mean is chosen, whereas the coefficients of $\gamma_2 = \gamma_3 = 0.2$ for global error, standard deviation and entropy are chosen. To avoid evaluating mechanisms with high utility dispersion and low utility mode values, a hard constraint is applied and only mechanisms that generate mean $\overline{\epsilon} < 0.1$ and standard deviation values $\sigma_\epsilon < 0.1$ are evaluated. The normalizing factors of Relations (5.16) and (5.14) are chosen after the application of this constraint.

A sensitivity analysis of the parameters for each masking mechanism is performed to evaluate the effect of different parameter values on the privacy and utility output of each masking mechanism. In the Laplace masking mechanism, increasing the scale parameter b of the distribution, also increases the total noise added to the dataset. In the sine mechanism, increasing the number and values of the coefficients, also increases the total generated noise. In Figure 5.3, a comparison of privacy and utility is shown between the two types of mechanisms. The values of utility and privacy are generated as shown in Section 5.5. The total noise is generated by measuring the noise level of each privacy setting on a sample of 100,000 sensor values[14]. The lines are smoothed by applying a moving average, to make the comparison clearer. For the same amount of total absolute generated noise $\sum_t |\psi_t|$, the Laplace privacy settings achieve higher privacy, often more than 1% over the sine polynomial privacy settings. The sine polynomial privacy settings achieve higher utility around 1% over the Laplace privacy settings. Therefore the results illustrated in Figure 5.2 are reflected in the privacy and utility values generated from the above parameterization. Moreover, the trade-off between privacy and utility is observable, as privacy increases with the decrease of utility and vice versa for both mechanisms.

Homogeneous System Evaluation

All the generated privacy settings are evaluated via the framework proposed in Section 5.3. The proposed framework filters out five privacy settings for five privacy bins of size 0.2. The constraint value for evaluating privacy settings is chosen empirically to be half of the bin size $\omega = 0.1$, to ensure low privacy dispersion, based on Relation (5.18). The resulting privacy settings are summarized in *Table* 5.1. The last two columns of the table illustrate the median privacy and utility values for each masking mechanism. The first column shows the id of each setting, which is used as reference in Figures *5.4 & 5.5*.

[14]This sample size is chosen to be large enough for statistical significance and small enough to reduce computation costs.

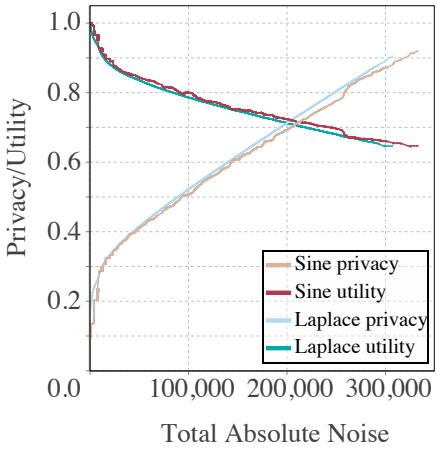


FIGURE 5.3: Comparison of sine polynomial and Laplace masking mechanisms in terms of privacy and utility.

TABLE 5.1: A table summarizing the performance of the five optimal privacy settings based on the parameters of the sine polynomial denoting the coefficient value for each factor of the polynomial or the scale value for a Laplace mechanism. In case of the sine polynomial, the first number from right is mapped to the first factor ($\xi = 1$) and so on.

ID	Masking	Parameters	Privacy	Utility
A	Sine	0.0-0.0-0.0-0.18-0.0	0.01	0.99
B	Laplace	0.005	0.20	0.98
C	Sine	0.6-0.6-0.0-0.9-0.3	0.40	0.84
D	Sine	1.2-0.3-0.6-1.2-0.9	0.60	0.76
E	Sine	1.5-1.5-1.2-0.3-1.2	0.80	0.68
N	none	-	0.00	1.00

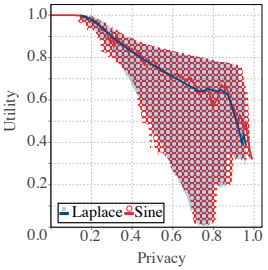
Figure 5.4a shows the generated privacy-utility values for all the privacy settings tested. Each color is mapped to the masking mechanism that is used to produce this setting. The line denotes the median value of utility at the given privacy value. The non-median privacy-utility values occur in the semi-transparent area. Upper and lower edges of the area denote the minimum and maximum utility value for the corresponding privacy value. Lower utility values for a given privacy point are generated from applications of the privacy setting on small subsets of the ECBT datasets, where $|N| \leq 1000$. The number of sensor values decreases with the number of users. Therefore, the noise cancellation is also reduced, as mentioned in Section 5.1. Hence, subsets with a lower number of sensor values produce lower utility values. The trade-off between privacy and utility is quantified, since the median curve and the edges of the surrounding area indicate a decrease in utility with the increase of privacy. In Figure 5.4b, the area of privacy-utility values of 5 privacy settings produced by the optimization process is shown in Section 5.4. Furthermore the "no masking" privacy setting is also considered, where users choose to use no privacy setting and send the values unmasked.

As it is shown, the privacy values of each privacy setting are within a range of lower than 0.2 privacy. The dispersion of utility increases for privacy settings that achieve higher privacy. The importance of offering higher rewards for the usage of higher utility mechanisms is validated, since high dispersion of utility is restrictive for accurate sum calculations by the data consumers. Figures 5.4c and 5.4d illustrate the privacy-utility trajectories for more than 1,000 users. It is evident that a data consumer can also increase utility and reduce its dispersion by attracting more users. Higher rewards in general, can also attract more users, so the utility dispersion is expected to decrease even more.

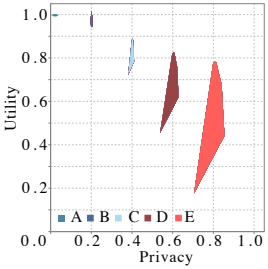
Heterogeneous System Evaluation

In heterogeneous system, the framework performance is evaluated under the use of different privacy settings from each user. The difference of privacy and utility between homogeneous and heterogeneous systems is quantified. This quantification is done by performing an exhaustive simulation. The simulation combines the ECBT dataset and the six privacy settings in *Table* 5.1. Every user of the ECBT dataset is assigned a privacy setting from *Table* 5.1. The percentage of users that are assigned each privacy setting is parameterized. A histogram with six bins is created. Each bin corresponds to the ID of a specific privacy setting from Table 5.1. The percentage assigned to a bin denotes the percentage of users using the respective privacy setting at this time point. To generate several possible scenarios for different distributions of user choices, the histogram is parameterized via a parameter sweep of all possible percentage values for each setting, with a step of 12.5%. This process produces over 1000 possible histograms. In figures 5.5a to 5.5d the heatmaps show the median and the interquantile range (IQR)[15] of privacy and utility for all histograms that the privacy setting has a higher percentage of users compared to the others. Such a

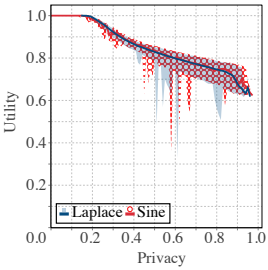
[15]IQR is considered a robust measure of scale, which is especially used for non-symmetric distributions. It measures the range between the 25^{th} and the 75^{th} quantiles.



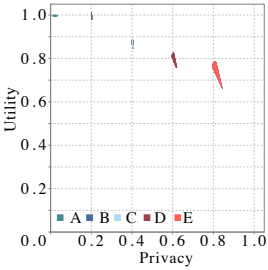
(a) Trajectory for all user set sizes



(b) Optimization results for all user set sizes



(c) Trajectory for all user sets with more than 1000 users



(d) Optimization results for user sets with more than 1000 users

FIGURE 5.4: Figures 5.4a & 5.4c show the privacy-utility trajectory of the privacy settings grouped by various masking mechanisms in the same color. Figures 5.4b & 5.4d illustrate the trajectories of the Pareto optimal privacy settings, which are generated by the proposed framework.

setting is referred to as dominant setting. This sorting of settings is done to examine the privacy-utility changes while users move from a higher to the next lower utility setting. The top row of the heatmap shows the homogeneous scenario case, where 100% of the users chose only one setting.

The analysis of the heatmap in Figure 5.5a shows an increase in privacy when the majority of users choose the more privacy-preserving settings of the homogeneous scenario. This effect is observed for any percentage of users for a dominant setting. A decrease in utility median is confirmed in 5.5c, when the majority of users shifts from less private to more private settings. A trade-off between privacy and utility is preserved in the heterogeneous scenario, regardless of the percentage of users that choose the dominant setting. Privacy values disperse more in heterogeneous systems, according to Figure 5.5b, as the percentage assigned to the dominant setting drops. The dispersion of privacy can reach up to 0.16, which is still lower than the bin range. In terms of utility, the dispersion is much lower on average. There is a dispersion

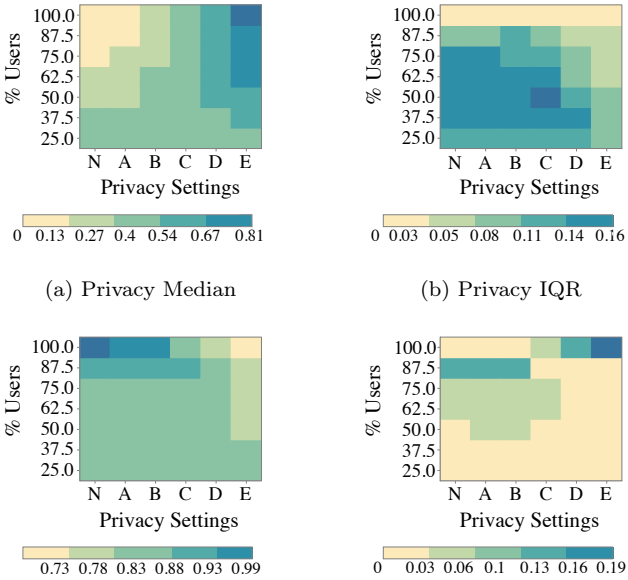
of around 0.1 for high utility mechanisms when they are dominant with 87.5% of users. A possible explanation for this is the reduction of noise cancellation of high privacy settings, due to the low percentage of users choosing them. Concluding, changing from a homogeneous system to a heterogeneous system preserves the trade-off between privacy and utility in the median values. Furthermore, the change to a heterogeneous system increases the dispersion of privacy-utility values for all the mechanisms, so the data consumer should expect the aggregates to be less accurate. Still, utility remains over 0.76 even if the IQR is subtracted from the median, indicating that the aggregate is still approximated even in the heterogeneous case. This validates empirically Theorem 1. In both cases it is efficient for the data consumer to shift user privacy choices to high utility mechanisms by offering them higher rewards. The randomness of the generated noise in a heterogeneous system does not create high variance or high expected global errors. Individual privacy is still preserved for all users and their privacy settings. The individual privacy value does not change between heterogeneous and homogeneous systems, since the privacy-setting choice of one user does not affect the added noise to the sensor values of the other ones.

5.6 Conclusion and Future Work

An optimization framework for the selection of privacy settings is introduced in this chapter. The framework computes privacy settings that maximize utility for different values of privacy. This framework can be utilized in privacy-preserving systems that calculate aggregation functions over privatized sensor data. The data producers of such system can self-determine the privacy setting of their choice, since it is guaranteed that it produces the desired privacy with very low deviation. For the data consumer of the system, it is guaranteed that, if the data producers are incentivized to use low-privacy settings and high utility settings, the approximated aggregate is highly accurate. Analytical as well as empirical evaluations using over $20,000$ privacy settings and real-world data from a Smart Grid pilot project confirm the viability of participatory data sharing under informational self-determination.

The outcomes of this chapter allow for privacy-preserving calculation of collective goals and constraints from individual values. Aggregating consumption values of individual consumers helps in calculating consumption totals and averages. These values can inform both individuals and shareholders about the current state of consumption. Furthermore, the privacy-preserving aggregated values can be used within objective functions and constraints of value-sensitive sustainability recommender systems. This capability is used in Chapter 4 to facilitate calculation of privacy-preserving collective constraints

For future work, the proposed framework can be improved by incorporating a machine learning process that computes personalized recommendations of privacy settings to each data producer by identifying the prior distribution of the sensor data and also the preferences of the data producer. Further empirical evaluations of framework can be performed by implementing other aggregation functions and using



(a) Privacy Median

(b) Privacy IQR

(c) Utility Median

(d) Utility IQR

FIGURE 5.5: Heatmaps showing the privacy and utility median and IQR values, for various distributions of privacy setting choices among the users. Each distribution of privacy settings represents a single experiment. The horizontal axis represents the dominant privacy setting in each experiment, while the vertical axis represents the number of users that adopted this setting. The colorscale represents the achieved privacy and utility median and IQR values. We observe that settings B,C and D achieve relatively low trade-offs between utility and privacy when they are dominant.

different datasets. Finally, an analytical proof that the sine polynomial additive noise is not colored and differentially private can be attempted.

Chapter

6

How Artificial Neural Networks Learn to Control Dynamical Systems

Even if a certain control policy is able to steer a system towards a target state, it may not be possible to implement it in practice because of resource and energy constraints. To determine admissible control policies, one typically resorts to optimal control (OC) methods that rely on cost functions that one wishes to minimize. Such cost functions may be used to minimize the strength and frequency of control signals or, more generally, the "control energy" [141]. In technical networks, energy has to be supplied to control the action of underlying electrical and mechanical components. In social and socio-economic networks [317], one can identify control energy with the resources or costs incurred (*e.g.*, economic and social costs of distancing policies) when changing the behavior of individual nodes [141]. For sustainability related applications, optimal control can be used to determine optimal carbon taxing policies [29] to reduce emissions without enforcing large taxes to impact the economy. Achieving optimal control of networked dynamical systems is thus a central task in control theory and its applications [318].

Recapping Chapter 2, the solution of general optimal control problems is based on two main approaches: (i) Pontryagin's maximum principle [319, 135] (necessary condition), which is a boundary-value problem in a Hamiltonian framework, or (ii) solving the Hamilton–Jacobi–Bellman (HJB) partial-differential equation (necessary and sufficient condition) [142]. Since the HJB equation usually does not admit smooth solutions [147], different approximate dynamic programming methods have been developed [320, 144].

To extend the above approaches (i) and (ii) to complex and analytically intractable systems, different methods relying on artificial neural networks (ANNs) have been used to represent certain functions appearing in the formulation of the corresponding control problems. One possibility is to use ANNs to obtain an approximate solution to the value function of the HJB equation [321, 320]. An alternative approach is based on the solution of Pontryagin's maximum principle via differentiable programming [322]. Control approaches that rely on Pontryagin's maximum principle explicitly account for a control energy term in the loss function and are based on the solutions of a system's evolution and co-state equations. In addition to relying on an energy regularization term in the loss function, the control framework of [322] is based on an extension of the maximum principle that includes higher-order

derivatives, requiring the underlying dynamical systems to be twice-differentiable. Differentiable programming has been applied to control systems with a maximum of 13 state variables [322], almost two orders of magnitude smaller than some of the high-dimensional dynamical systems we study in this work. Recent advances in automatic differentiation and physics-informed ANNs [323] also contributed to the further development of modeling and control approaches. Physics-informed neural networks use Lagrangian- and Hamiltonian-based formulations of physicals models as priors for different learning tasks [324, 325, 326]. They are useful tools to model partially unknown systems [324] and have been also applied to control tasks [325, 326].

Contrary to the above approaches, we show that it is possible to generate control signals that resemble those of optimal control [141] without reyling on and solving maximum principle or HJB equations. To do so, we present a time-depended variant of NODEC, termed AI Pontryagin, an ANN that overcomes several limitations of traditional optimal control methods resulting from the analytical and computational intractability of many complex and high-dimensional control tasks. AI Pontryagin extends neural ordinary differential equations (ODEs) [136] to general control problems, and efficiently steers complex dynamical systems towards desired target states by learning control trajectories that resemble those obtained with optimal control methods. It does so by exploring the vector field of the underlying dynamical system in a "go-with-the-flow" manner, without explicitly accounting for an energy-regularization term in the corresponding loss function. That is, AI Pontryagin minimizes the control energy [141] without evaluating an energy cost functional, leading to a substantially improved performance compared to existing control frameworks. Using analytical and numerical arguments, we show why AI Pontryagin is able to closely resemble the control energy of optimal control through an implicit energy regularization, resulting from the interplay of ANN initialization and an induced gradient descent. Applications of such frameworks to feedback control and a comparison with reinforcement learning are provided in Chapter 2.

6.1 Results

Controlling Dynamical Systems with AI Pontryagin

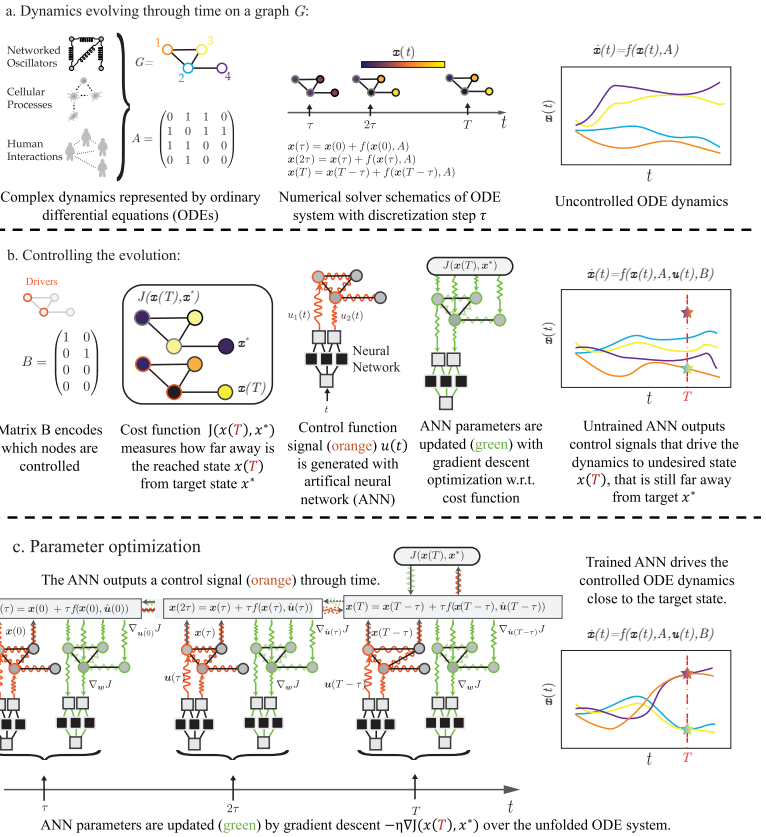


FIGURE 6.1: **Overview of AI Pontryagin. (a)** Illustration of a complex, uncontrolled dynamical system represented by ODEs, and discretization steps of numerical solver. **(b)** Overview of basic elements necessary to control complex systems with ANNs. **(c)** Gradient descent training of ANN parameters over the unfolded ODE system.

Before introducing the basic principles of AI Pontryagin, we first provide a mathematical formulation of the control problem of networked dynamical systems. We consider a network that consists of N nodes whose states are represented by the state vector $\boldsymbol{x}(t) \in \mathbb{R}^N$. Initially, nodes are in state $\boldsymbol{x}(0)$ and steered towards a target state $\boldsymbol{x}^*$ at time T (*i.e.*, $\boldsymbol{x}(T) = \boldsymbol{x}^*$) by means of suitable control inputs.

Interactions between nodes are described by the dynamical system

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{6.1}$$

and are subject to the constraint that the control function $\boldsymbol{u}(t) \in \mathbb{R}^M$ minimizes the cost function

$$J = \int_0^T L(\boldsymbol{x}(t'), \boldsymbol{u}(t'))\, dt' + C(\boldsymbol{x}(T))\,. \tag{6.2}$$

The function $f\colon \mathbb{R}^N \to \mathbb{R}^N$ in Relation (6.1) accounts for both the interactions between nodes $1, \ldots, N$ and the influence of external control inputs $\boldsymbol{u}(t)$ on the dynamics. Note that the number of control inputs M is smaller than or equal to N. For linear systems, we describe node-node interactions and external control inputs by $f(\boldsymbol{x}, \boldsymbol{u}) = A\boldsymbol{x} + B\boldsymbol{u}$. The first term in Relation (6.2) is the integrated cost over the control horizon T, *e.g.*, the control energy

$$E_T[\boldsymbol{u}] = \int_0^T \left\|\boldsymbol{u}(t')\right\|_2^2 dt' \tag{6.3}$$

if $L = \|\boldsymbol{u}(t')\|_2^2$. $C(\boldsymbol{x}(T))$ is the final cost (or bequest value). Most common formulations of optimal control include the control-energy term Relation (6.3) directly in the cost function [141, 327]. This approach corresponds to an explicit minimization of the control energy.

With AI Pontryagin we take a complementary approach to reach a desired target state $\boldsymbol{x}^*$ in finite time T and proceed in two steps. First, we approximate and solve the dynamical system in terms of neural ODEs [136]. In particular, we describe the control input $\boldsymbol{u}(t)$ by an ANN with weight vector $\boldsymbol{w}$ such that the corresponding control-input representation is $\hat{\boldsymbol{u}}(t; \boldsymbol{w})$. Second, we use a suitable loss function $J(\boldsymbol{x}, \boldsymbol{x}^*)$ and a gradient-descent algorithm to iteratively determine the weight vector $\boldsymbol{w}$ according to

$$\boldsymbol{w}^{(n+1)} = \boldsymbol{w}^{(n)} - \eta \nabla_{\boldsymbol{w}^{(n)}} J(\boldsymbol{x}, \boldsymbol{x}^*), \tag{6.4}$$

where the superscript indicates the current number of gradient-descent steps, and η is the learning rate. For the loss function $J(\cdot)$, we use the mean-squared error

$$J(\boldsymbol{x}(T), \boldsymbol{x}^*) = \frac{1}{N} \left\|\boldsymbol{x}(T) - \boldsymbol{x}^*\right\|_2^2. \tag{6.5}$$

In order to calculate $\nabla_{\boldsymbol{w}^{(n)}} J(\cdot)$, we use automatic differentiation methods [328], where the gradients "flow" through the underlying ANN, which is time-unfolded [168] by ODE solvers [158]. We show a schematic of the forward and backward passes of AI Pontryagin and its coupling to an underlying dynamical system in Figure 6.1.

In the following paragraphs, we will show that AI Pontryagin approximates optimal control by minimizing the control energy Relation (6.3), without including the energy cost Relation (6.3) in the loss function Relation (6.5) and without having any prior knowledge on the structure of optimal control signals. All neural-network architectures, hyperparameters, and numerical solvers are reported in the Methods.

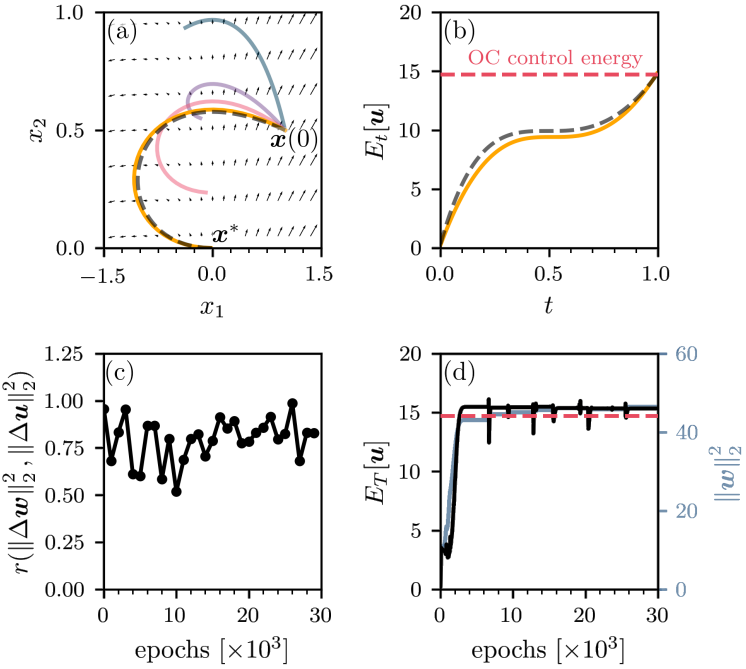


FIGURE 6.2: **Controlling a two-node system with AI Pontryagin. (a)** Different AI Pontyagin control trajectories for $\boldsymbol{x}(0) = (1, 0.5)^{\mathrm{T}}$, $\boldsymbol{x}^* = (0, 0)^{\mathrm{T}}$, and $T = 1$ after 500 (blue), 1,500 (purple), 2,000 (red), and 30,000 (orange) training epochs with learning rate $\eta = 0.02$. The dashed black line is the corresponding optimal control trajectory and black arrows indicate the vector field of the linear dynamical system $f(\boldsymbol{x}, \boldsymbol{u}) = A\boldsymbol{x} + B\boldsymbol{u}$ with matrices A and B as in Relation (6.8) **(b)** Evolution of the control energy $E_t[\boldsymbol{u}]$ for AI Pontryagin after 30,000 training epochs (solid orange line) and optimal control (dashed black line). **(c)** Correlations between squared norm differences of ANN weights $\mathbf{w}$ and control inputs $\mathbf{u}$. **(d)** The control energy $E_T[\boldsymbol{u}]$ (black solid line) and squared norm of the ANN weights $\mathbf{w}$ (solid grey line) of AI Pontryagin as a function of training epochs. In (b,d), we indicate the total OC control energy by a dashed red line.

Approximating Optimal Control

We now compare the control performance of AI Pontryagin for linear systems (*i.e.*, $f(\boldsymbol{x}, \boldsymbol{u}) = A\boldsymbol{x} + B\boldsymbol{u}$), for which there exist analytical OC inputs [141]

$$\boldsymbol{u}^*(t) = B^\top e^{A^\top (T-t)} W(T)^{-1} \boldsymbol{v}(T) \tag{6.6}$$

that minimize the control energy $E[\boldsymbol{u}(t)]$ (Relation (6.3)). For the derivation of Relation (6.6), one applies Pontryagin's maximum principle to the Hamiltonian $H = \|\boldsymbol{u}(t)\|_2^2 + \lambda^T (A\boldsymbol{x} + B\boldsymbol{u})$ [141], where λ is an adjoint variable. The vector $\boldsymbol{v}(T) = \boldsymbol{x}^* - e^{AT}\boldsymbol{x}_0$ in Relation (6.6) is the difference between the target state $\boldsymbol{x}^*$ and reached stateat T from $\boldsymbol{x}(0)$ under free evolution. The matrix $W(T)$ is the controllability Gramian and defined as

$$W(T) = \int_0^T e^{At} B B^\top e^{A^\top t} \, dt. \tag{6.7}$$

As an example of linear dynamics, we consider a two-state system with [141, 327]

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{6.8}$$

The control task is to steer the system from $\boldsymbol{x}(0) = (1, 0.5)^\mathrm{T}$ to $\boldsymbol{x}^* = (0, 0)^\mathrm{T}$ in finite time $T = 1$.

In Figure 6.2(a), we show AI Pontryagin trajectories after 500 (blue), 1,000 (purple), 1,500 (red), and 30,000 (orange) training epochs together with an OC control trajectory (dashed black line). Note that the geodesic that connects $\boldsymbol{x}(0)$ and $\boldsymbol{x}^*(T)$ is not minimizing the control energy, because it would require large control inputs to steer the dynamics against the vector field (black arrows in Figure 6.2(a)). In alignment with the almost identical control trajectories of AI Pontryagin and OC, we also find that the energy evolution of AI Pontryagin almost perfectly coincides with that of OC (see Figure 6.2(b)), hinting at an implicit energy regularization of AI Pontryagin.

Implicit Energy Regularization

To provide insights into the observed implicit energy regularization of AI Pontryagin (see Figure 6.2(b)), we show that a gradient descent in the ANN weights $\boldsymbol{w}$ induces a gradient descent in the control input $\hat{\boldsymbol{u}}(t; \boldsymbol{w})$.

The evolution of the state vector $\boldsymbol{x}(t)$ is described by Relation (6.1) and is a function of $\hat{\boldsymbol{u}}(t; \boldsymbol{w})$. We now expand $\hat{\boldsymbol{u}}(t; \boldsymbol{w}^{(n+1)}) = \hat{\boldsymbol{u}}(t; \boldsymbol{w}^{(n)} + \Delta\boldsymbol{w}^{(n)})$ with $\Delta\boldsymbol{w}^{(n)} = -\eta \nabla_{\boldsymbol{w}^{(n)}} J$ for small $\Delta\boldsymbol{w}^{(n)}$ while keeping t constant. This expansion yields

$$\hat{\boldsymbol{u}}(t; \boldsymbol{w}^{(n+1)}) = \hat{\boldsymbol{u}}(t; \boldsymbol{w}^{(n)}) + \mathcal{J}_{\hat{\boldsymbol{u}}} \Delta\boldsymbol{w}^{(n)}, \tag{6.9}$$

where $\mathcal{J}_{\hat{\boldsymbol{u}}}$ is the Jacobian of $\hat{\boldsymbol{u}}$ with elements $(\mathcal{J}_{\hat{\boldsymbol{u}}})_{ij} = \partial \hat{\boldsymbol{u}}_i / \partial \boldsymbol{w}_j$. Note that we can make $\Delta\boldsymbol{w}^{(n)}$ arbitrarily small by using a small learning rate η.

Since $\Delta \boldsymbol{w}^{(n)} \propto \nabla_{\boldsymbol{w}^{(n)}} J$ and $\nabla_{\boldsymbol{w}^{(n)}} J = \mathcal{J}_{\hat{\boldsymbol{u}}}^T \nabla_{\hat{\boldsymbol{u}}} J$, we obtain

$$\hat{\boldsymbol{u}}(t; \boldsymbol{w}^{(n+1)}) = \hat{\boldsymbol{u}}(t; \boldsymbol{w}^{(n)}) - \eta \mathcal{J}_{\hat{\boldsymbol{u}}} \mathcal{J}_{\hat{\boldsymbol{u}}}^T \nabla_{\hat{\boldsymbol{u}}} J. \tag{6.10}$$

According to Relation (6.10), a gradient descent in $\boldsymbol{w}$ (Relation (6.4)) may induce a gradient descent in $\hat{\boldsymbol{u}}$, where the square matrix $\mathcal{J}_{\hat{\boldsymbol{u}}} \mathcal{J}_{\hat{\boldsymbol{u}}}^T$ acts as a linear transformation on $\nabla_{\hat{\boldsymbol{u}}} J$.

To better understand the implications of this result, we briefly summarize the control steps of AI Pontryagin. As described in the prior paragraphs and as illustrated in Figure 6.2(a), AI Pontryagin starts with a small initial control signal $\hat{\boldsymbol{u}}^{(0)}(t; \boldsymbol{w}^{(0)})$, then integrates the dynamical system Relation (6.1), and performs a gradient descent in $\boldsymbol{w}$ according to Relation (6.4). The closer the final state $\boldsymbol{x}(T)$ to the target state $\boldsymbol{x}^*$, the smaller the loss Relation (6.5) and the change in $\boldsymbol{w}$ (and in $\hat{\boldsymbol{u}}$ due to Relation (6.10)). If we initialize AI Pontryagin with a sufficiently small control input and learning rate, it will produce control trajectories that follow the vector field of the dynamical system in a "go-with-the-flow" manner and slowly adapt $\hat{\boldsymbol{u}}$ to reach the desired target state. Because of the induced gradient descent Relation (6.10), the resulting control approximates OC methods that minimize the control energy (see the comparison between the final control energy of OC and AI Pontryagin in Figure 6.2(b,d)). This way of controlling dynamical systems is markedly different from standard (optimal) control formulations [329] that are, for instance, based on Pontryagin's maximum principle and require one to explicitly minimize the control energy by including $\|\boldsymbol{u}\|_2^2$ in the Hamiltonian and solving the adjoint system [329]. AI Pontryagin thus provides a complementary approach for solving general control problems.

The induced gradient descent Relation (6.10) can be directly observed in the positive correlations between $\|\Delta \boldsymbol{w}\|_2^2 = \left\| \boldsymbol{w}^{(n+1)} - \boldsymbol{w}^{(n)} \right\|_2^2$ and $\|\Delta \boldsymbol{u}\|_2^2 = \left\| \boldsymbol{u}^{(n+1)} - \boldsymbol{u}^{(n)} \right\|_2^2$ (see Figure 6.2(c)). Black disks indicate correlation coefficients ($p < 10^{-9}$) that are each calculated for 10^3 consecutive epochs and solid black lines are guides to the eye. After initializing AI Pontryagin for the linear two-state system Relation (6.8) with weights that correspond to a small control input, we observe positive correlations between $\|\Delta \boldsymbol{w}\|_2^2$ and $\|\Delta \boldsymbol{u}\|_2^2$ with a large correlation coefficient of 0.96 for the first 1,000 training epochs. The mean correlation coefficient is about 0.76. Changes in the correlation behavior reflect different training stages that are necessary to capture the strong curvature in the OC control trajectory (dashed black line in Figure 6.2(a)). Between 1,500 and 2,000 training epochs, AI Pontryagin approximates the basic shape of the OC trajectory (solid red line and red disks in Figure 6.2(a)) and then fine-tunes the weights $\boldsymbol{w}$ to match OC as closely as possible (solid orange line and orange disks in Figure 6.2(a)). The initial OC approximation phase that lasts up to about 2,000 training epochs (before weight "fine-tuning") is also visible in the evolution of $\|\boldsymbol{w}\|_2^2$ and $\|\boldsymbol{u}\|_2^2$ (see Figure 6.2(d)).

We again emphasize that the performance of AI Pontryagin and its induced-gradient descent mechanism depends on the choice of initial weights $\boldsymbol{w}^0$ (and thus $\boldsymbol{u}(t; \boldsymbol{w}^0)$). The initialization that we use to obtain the results of Figure 6.2 is based on energy values which are distributed in the interval $[5, 7]$. These values are small

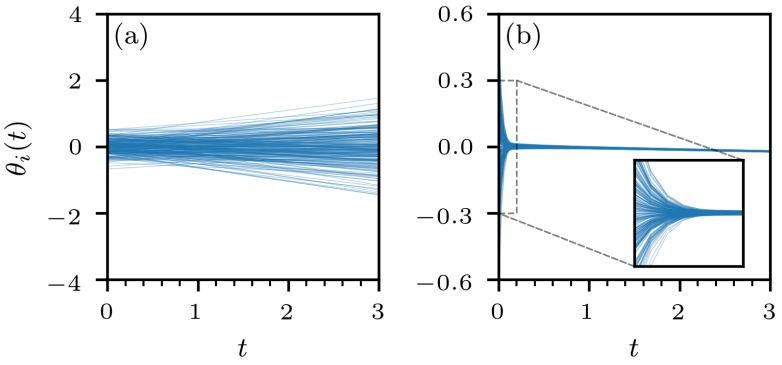


FIGURE 6.3: **Synchronization of coupled oscillators.** The evolution of oscillator phases $\theta_i(t)$ $(1 \leq i \leq N)$ in a complete network that consists of $N = 225$ coupled Kuramoto oscillators (Relations (6.11) and (6.12)) with a subcritical coupling constant $K = 0.1K^*$, which does not lead to synchronicity
. All phases are initially distributed according to a normal distribution with mean 0 and standard deviation 0.2. **(a)** The control input is set to $u_i(t) = 1$ for all i ("uncontrolled dynamics"), leading to increasing phase differences over time. **(b)** AI Pontryagin synchronizes the system of coupled oscillators.

enough for AI Pontryagin to let it explore the vector field of the underlying dynamical system and approximate OC.

In the Appendix Chapter E, we provide additional results on the ability of AI Pontryagin to control dynamics on directed networks. We show that AI Pontryagin is able to produce control signals with a control energy resembling that of the corresponding OC solution, which we verify by calculating the corresponding optimal control signals if possible. In the Appendix, we also study the robustness of AI Pontryagin-based control with respect to different noise levels in the observed reached state.

After having outlined the mechanisms underlying the observed energy regularization of AI Pontryagin, we now turn towards non-linear systems.

AI Pontryagin Control of Kuramoto Oscillators

As an example of a non-linear system, we consider the Kuramoto model [191], which describes coupled oscillators with phases θ_i and intrinsic frequencies ω_i $(1 \leq i \leq N)$ according to [191]

$$\begin{aligned} \dot{\Theta}(t) &= \Omega + f(\Theta(t), u(t)), \\ \Theta(0) &= \Theta_0, \end{aligned} \tag{6.11}$$

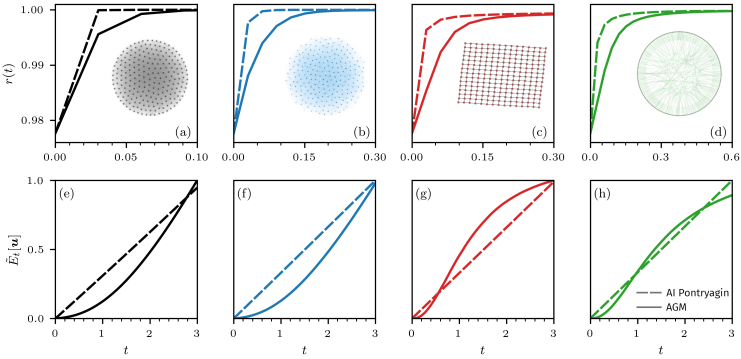


FIGURE 6.4: **Controlling coupled oscillators with AI Pontryagin and the AGM.** We test the performance of AI Pontryagin and AGM (Relations (6.17) and (6.18)) to control coupled Kuramoto oscillators on a **(a,e)** complete network (black lines), **(b,f)** Erdős–Rényi network $G(N,p)$ with $p=0.3$ (blue lines), **(c,g)** square lattice (red lines), and **(d,h)** Watts–Strogatz network with degree $k=5$ and rewiring probability $p=0.3$ (green lines). All graphs have $N=225$ nodes and the total simulation time is $T=3$. Panels (a–d) show the order parameter $r(t)$ and panels (e–h) show the control energy $\tilde{E}_t(\mathbf{u}) = E_t(\mathbf{u})/\max(E_t^{AIP}(\mathbf{u}), E_t^{AGM}(\mathbf{u}))$. Dashed and solid lines indicate AI Pontryagin and AGM solutions, respectively.

where $\Theta = (\theta_1, \ldots, \theta_N)^\top$ and $\Omega = (\omega_1, \ldots, \omega_N)^\top$. In our following numerical experiments, we use natural frequencies and initial phases that are normally-distributed with mean 0 and standard deviation 0.2. Interactions between oscillators and the influence of control inputs $u_i(t)$ on oscillator i are modeled via

$$f_i(\Theta(t), u(t)) = \frac{K u_i(t)}{N} \sum_{j=1}^{N} A_{ij} \sin(\theta_j(t) - \theta_i(t)), \tag{6.12}$$

where K is the coupling strength and A_{ij} are the adjacency matrix components of the underlying (undirected) network. As a measure of synchronization at the final time T, we use the complete synchronization condition [330, 150]

$$|\dot{\theta}_i(T) - \dot{\theta}_j(T)| = 0 \text{ for } (i,j) \in E, \tag{6.13}$$

where E is the set of edges. If Relation (6.13) is satisfied, all connected oscillators have constant phase differences. For control inputs that are equal to 1 (*i.e.*, $u_i(t) = 1$ for all i), the system Relation (6.11) has a unique and stable synchronized state if the coupling constant exceeds a critical value [331]

$$K^* = \left\| L^\dagger \Omega \right\|_{E,\infty}, \tag{6.14}$$

where $L^\dagger$ is the pseudo-inverse of the corresponding graph Laplacian and $\|\boldsymbol{x}\|_{E,\infty} = \max_{(i,j)\in E} |x_i - x_j|$ is the maximum distance between elements in $\boldsymbol{x} = (x_1, \ldots, x_N)^\top$

that are connected via an edge in E. In all numerical simulations, we use a subcritical coupling constant $K = 0.1K^*$ such that control inputs $u_i(t) > 1$ are needed to synchronize the system.

For a global control $u(t)$ (*i.e.*, $u_i(t) = u(t)$ for all i), there exists an OC input $u^*(t)$ satisfying [150]

$$u^* = \min_u J(\Theta(T), u) \tag{6.15}$$

$$J(\Theta(T), u) = \frac{1}{2} \sum_{i,j} A_{ij} \sin^2(\theta_j(T) - \theta_i(T)) + \frac{\beta}{2} E[u], \tag{6.16}$$

where the parameter β determines the influence of the energy regularization term. Note that minimizing $J(\Theta, u)$ is consistent with Relation (6.13).

An optimal control for the outlined non-linear control problem, the so-called the adjoint-gradient method (AGM), can be derived using Pontryagin's maximum principle and a gradient descent in u [150]:

$$u^{(n+1)} = u^{(n)} - \tilde{\eta} \left[\beta u^{(n)} + \frac{K}{N} \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{N} A_{ij} \sin(\theta_j - \theta_i) \right], \tag{6.17}$$

where $\tilde{\eta}$ is the AGM learning rate and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^\top$ is the solution of the adjoint system

$$\begin{aligned} -\dot{\lambda}_i = &-\frac{Ku\lambda_i}{N} \sum_{i \neq j} A_{ij} \cos(\theta_j - \theta_i) \\ &+ \frac{Ku}{N} \sum_{i \neq j} A_{ij} \lambda_j \cos(\theta_j - \theta_i), \end{aligned} \tag{6.18}$$

with $\lambda_i(T) = 1/2 \sum_{i \neq j} A_{ij} \sin(2\theta_i(T) - 2\theta_j(T))$.

We compare the control performance of AI Pontryagin, which solves Relation (6.11) using neural ODEs, with that of the AGM for a global control function. AI Pontryagin directly learns $\hat{u}^*(t; \boldsymbol{w})$ based on the loss function Relation (6.16) with a gradient descent in $\boldsymbol{w}$ and *without* energy regularization term $\beta E[u]/2$. We denote this loss function by

$$J_1(\Theta(T)) = \frac{1}{2} \sum_{i,j} A_{ij} \sin^2(\theta_j(T) - \theta_i(T)) \,. \tag{6.19}$$

The learning rates η (Relation (6.4)) and $\tilde{\eta}$ (Relation (6.17)) are chosen such that the ratio of the order parameter values of both control methods is approximately 1. We discuss in the Appendix that a high degree of synchronization can be obtained by controlling a fraction of all nodes and we show how a maximum matching approach [129] can be used to determine driver nodes for controlling linear dynamics with more than 1,000 nodes. All employed network architectures and training parameters are summarized in the Methods and in our online code repository `https://github.com/asikist/nnc`.

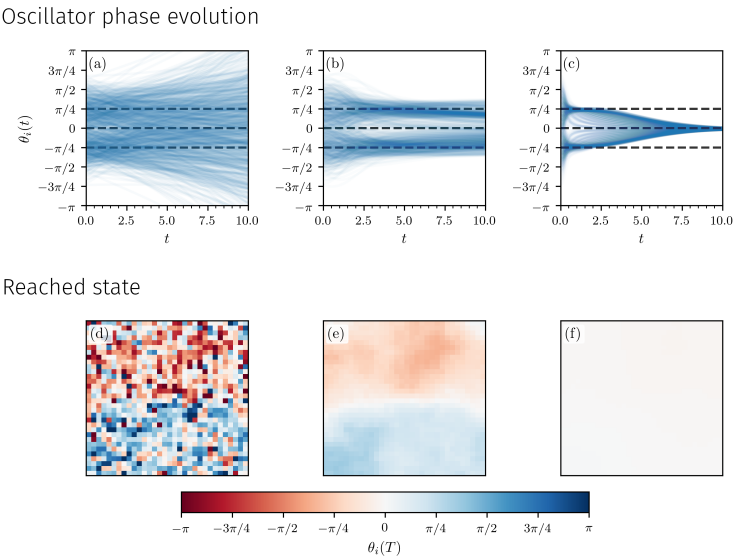


FIGURE 6.5: **Kuramoto dynamics and different target states.** The evolution of the phases $\theta_i(t)$ $(1 \leq i \leq N)$ of $N = 1,024$ coupled Kuramoto oscillators (Relations (6.11) and (6.12)) that are arranged in a square lattice without periodic boundary conditions. We use a subcritical coupling constant $K = 0.01K^*$ and set $T = 10$. In all simulations, oscillator phases are initially distributed according to a bimodal Gaussian distribution with means $-\pi/4, \pi/4$ and variance 0.5. The top panels show the evolution of $\theta_i(t)$ and the bottom panels show the spatial distribution of oscillator phases $\theta_i(T)$ at time $T = 10$. Each pixel in the 32×32 bottom panels represents the phase value of one specific oscillator in the reached state. **(a,d)** The control input is set to $u_i(t) = 1$ for all i ("uncontrolled dynamics"), leading to increasing phase differences over time. **(b,e)** AI Pontryagin steers oscillator phases towards $-\pi/4$ and $\pi/4$ by minimizing the loss function $J_2(\Theta(T))$ (Relation (6.21)). **(c,f)** AI Pontryagin synchronizes the system of coupled oscillators by minimizing the loss function $J_1(\Theta(T))$ (Relation (6.19)). The dashed black lines in (a–c) are guides-to-the-eye indicating phases with values $-\pi/4$, 0, and $\pi/4$. Learning rates varied between 10^{-2}–10^{1} in (b,e) and 0.1 in (c,f).

For a complete graph with $N = 225$ nodes and $T = 3$, we show a system of uncontrolled oscillators with $u_i(t) = 1$ for all i in Figure 6.3(a). As shown in Figure 6.3(b), AI Pontryagin can learn control inputs that drive the system of coupled oscillators into a synchronized state. To quantify the degree of synchronization, we

use the order parameter [1]

$$r(t) = N^{-1} \sqrt{\sum_{i,j} \cos\left[\theta_j(t) - \theta_i(t)\right]}\,. \tag{6.20}$$

A value of $r(t) = 1$ indicates that all oscillators have the same phase.

In Figure 6.4 we show the evolution of the order parameter $r(t)$ and control energy $E_t[\mathbf{u}]$ for AGM (solid lines) and AI Pontryagin (dashed lines). We study the control performance of both methods on a complete graph (black lines), Erdős–Rényi network $G(N, p)$ with $p = 0.3$ (blue lines), square lattice (red lines), and Watts–Strogatz network with degree $k = 5$ and rewiring probability $p = 0.3$ (green lines). All networks consist of $N = 225$ oscillators.

For all studied networks, we observe that AI Pontryagin reaches synchronization slightly faster than the AGM (see Figure 6.4(a–d)). We optimized the hyperparameters (*e.g.*, the number of training epochs) of the ANN underlying AI Pontryagin such that the control energy and degree of synchronization lie in a similar range to those of the AGM (see Figure 6.4(e–h)). Our results thus indicate that AI Pontryagin is able to achieve control energies similar to those of OC also for non-linear networked dynamics.

Next, we demonstrate the ability of AI Pontryagin to control oscillator systems with a number of nodes that is about one order of magnitude larger than that considered in Figure 6.4. We again compare AI Pontryagin with the AGM and set $T = 0.5$. Numerical experiments are performed on a square lattice with $N = 2,500$ coupled oscillators, and we find that the control energy and order parameter ratios are $E_T^{\mathrm{AI}P}(\mathbf{u})/E_T^{\mathrm{A}GM}(\mathbf{u}) \approx 1.0045$ and $r(T)^{\mathrm{AI}P}/r(T)^{\mathrm{A}GM} \approx 0.9999$, respectively. AI Pontryagin and the AGM reach similar order parameter and control energy values at time $T = 0.5$, indicating that both methods are able to control the larger-scale oscillator system.

For a runtime performance comparison, we also measure the learning time (or wall-clock time) associated with controlling the larger-scale oscillator system. To do so, we determine the runtime of 50 AGM and AI Pontryagin control realizations. The mean runtimes are 74 s and 1.03 s for the AGM and AI Pontryagin, respectively. For the studied oscillator system, the training time of AI Pontryagin is thus about two orders of magnitude smaller than that of the AGM. In the Appendix, we analyze the differences in runtime between AI Pontryagin and the AGM in more detail. To identify the main computational bottlenecks in the AGM, we performed a detailed runtime analysis of all code segments and found that the adjoint system solver

[1] Here, we used that the square of the magnitude of the complex order parameter $z = re^{i\psi(t)} = N^{-1}\sum_{j=1}^{N} e^{i\theta_j(t)}$ [191] can be expressed as

$$\begin{aligned} r(t)^2 = |z|^2 &= N^{-2} \sum_{i,j} e^{i(\theta_j(t) - \theta_i(t))} \\ &= N^{-2} \sum_{i,j} \cos\left[\theta_j(t) - \theta_i(t)\right]\,. \end{aligned}$$

requires very small step sizes to resolve the interaction between the adjoint system (Relation (6.18)) and the gradient descent (Relation (6.17)) in the control functions.

n a final numerical experiment, we show that AI Pontryagin is able to steer coupled Kuramoto oscillators to a target state that is different from the fully synchronized one. As an example of such a target state, we consider the control target to steer oscillators either towards $-\pi/4$ or $\pi/4$. This control goal can be described by the loss function

$$J_2(\Theta(T)) = \frac{1}{2}\sum_{i=1}^{N}\left(|\theta_i(T)| - \frac{\pi}{4}\right)^2 . \tag{6.21}$$

We show in Figure 6.5 that AI Pontryagin can use the loss function Relation (6.21) to reach target states in a lattice where oscillators are arranged in two spatially separated groups, each consisting of oscillators with phase values $\theta_i(T)$ $(1 \leq i \leq N)$ that are approximately $-\pi/4$ or $\pi/4$. In Figure 6.5(a,d), we do not control the square lattice consisting of $N = 1,024$ coupled Kuramoto oscillators with a subcritical coupling constant $K = 0.01K^*$, and we observe increasing phase differences over time. Using the loss function $J_2(\Theta(T))$ shows that AI Pontryagin can steer the system of coupled oscillators towards $-\pi/4$ and $\pi/4$ at the control time T (see Figure 6.5(b,e)). Oscillators that are located in the upper half of the square lattice in Figure 6.5(e) reached phase values close to $-\pi/4$ (indicated by light orange pixels), while oscillators in the lower half of the same lattice reached phase values of approximately $\pi/4$ (indicated by light blue pixels). By employing the loss function $J_1(\Theta(T))$, we can also use AI Pontryagin to synchronize the same system of coupled oscillators (dashed black line in Figure 6.5(c,f)). These results show that AI Pontryagin can be used in conjunction with different loss functions.

To summarize, AI Pontryagin has two key advantages over traditional adjoint-system-based control methods. First, approximate optimal control trajectories can be obtained without deriving and solving the adjoint system. The only inputs necessary are (i) the dynamical system, (ii) its initial state, and (iii) the desired target state. Second, the runtime of AI Pontryagin may be substantially faster than that of adjoint-gradient methods.

6.2 Discussion

The optimal control of networked dynamical systems is associated with minimizing a certain cost function, *e.g.*, the strength and frequency of a control signal, or, more generally, the control energy Relation (6.2). Traditional control approaches such as Pontryagin's maximum principle or the HJB equation are often analytically and computationally intractable when applied to complex dynamical systems.

In this work, we demonstrated the ability of AI Pontryagin, a control framework that is based on neural ODEs, to steer linear and non-linear networked dynamical systems into desired target states. AI Pontryagin uses as inputs the underlying dynamical system, and initial and target states. For the considered linear dynamics, we compared AI Pontryagin with corresponding analytical optimal control solutions

and found that AI Pontryagin is not only able to drive undirected and directed complex networks of dynamical systems into desired target states, but also is able to approximate the optimal control energy. We supported this observation with analytical arguments and further compared AI Pontryagin with an optimal control method for synchronizing oscillators in different networks, again showing that AI Pontryagin is able to approximate the optimal control energy.

AI Pontryagin is a very versatile control framework that complements existing optimal control approaches, and solves high-dimensional and analytically intractable control problems. Finally, there are various interesting avenues for further research. One possible direction for future work is the application of AI Pontryagin to solve complex quantum control problems to enhance robust performance of quantum systems [332]. Another possible direction for future research is to study the ability of AI Pontryagin to calculate optimal controls that preserve generator synchronicity during cascading failures, and ultimately avoid blackouts [152, 83]. For such complex control tasks, it may be useful to combine physics-informed neural networks, such as those studied in [324], with our neural-network-based control approach to learn and control the dynamics of partially unknown systems.

6.3 Methods

Both algorithms, AI Pontryagin and the AGM, are implemented in `pytorch`.

All ANNs that we use to represent the control input $\hat{\boldsymbol{u}}(t;\boldsymbol{w})$ in AI Pontryagin take the time t as an input. To numerically integrate the studied dynamical systems, we apply the Dormand–Prince (DOPRI) method with adaptive step size during training and evaluation [333].

In the following paragraphs, we summarize the ANN architectures and hyperparameters that we used in our numerical experiments.

Two-State System. The ANN that we use to control the two-state system Relation (6.8) consists of a single hidden layer with 6 neurons and an exponential linear unit (ELU) activation. We transform the hidden layer output to the control signal via a linear layer with 1 neuron that describes the single control input in Relation (6.8). We initialize the ANN weights $\boldsymbol{w}$ with the Kaiming uniform initialization algorithm [334]. For the gradient descent in $\boldsymbol{w}$ (Relation (6.4)), we use a learning rate $\eta = 0.02$. The number of time steps is 40.

Kuramoto Model. The graph properties and ANN hyperparameters for controlling Kuramoto dynamics are summarized in Table 6.2. Independent of the underlying graph, we use the same number of hidden layers, hidden layer neurons, and training epochs for the numerical experiments in Figure 6.4. The activation function (ELU) is also the same in all numerical experiments. The number of time steps is 100. For the runtime comparison between AI Pontryagin and the AGM, we use the command `timeit` in `python`. In accordance with [150], the energy regularization parameter β of the AGM is set to 10^{-7} (see the Appendix for a more detailed analysis of the

Graph	Edges (undirected)	Nodes	Learning rate η	Learning rate $\tilde{\eta}$
Complete	25,200	225	0.4	5
Erdős–Rényi	7,569	225	0.4	5
Square lattice	420	225	0.32	25
Watts–Strogatz	450	225	0.31	15
Square lattice	4,900	2,500	0.0125	0.5

TABLE 6.2: Learning rates η (AI Pontryagin) and $\tilde{\eta}$ (AGM) used for learning the control of Kuramoto dynamics on different graphs. All ANNs use stochastic gradient descent for learning and only differ in their learning rate. The number of hidden layers and hidden layer neurons are 1 and 2, respectively. We use an ELU activation function and train the ANN for two epochs. At each node, we include a bias term and set all weights initially to a value of 10^{-3}.

AGM control performance on β). Initially, we set all ANN weights to a value of 10^{-3}. For the second numerical experiment involving the loss function Relation (6.21), we use the the Kaiman initialization method [334] and 16 ELU activations in one hidden layer.

Chapter

7

Conclusion

Would a different decision help to reach our personal and collective goals with less costs? Would another goal allow for better decisions and the ability to pursue more important goals today? Do the goals and our decisions lead to a brighter future or catastrophe? In the previous chapters, we observe several quantitative and qualitative arguments regarding the challenging nature of these questions. We need to use all the technological tools available to us to answer these questions and avoid potential environmental disasters and societal inequalities. Based on the observed results, AI is a powerful and useful tool to support us in dealing with sustainable decision-making. Nevertheless, it comes with several potential caveats.

To address possible shortcomings of extensive AI use, such as privacy intrusion, loss of autonomy, and rise of inequality, the thesis follows the paradigm of value-sensitive AI. The computational capabilities of AI are enhanced with constraints, objectives, and other design components that represent and incorporate personal values. In doing so, AI uses are extended to support individuals in sustainable decision-making that respects their personal values. In the meantime, value-sensitive AI allows both individuals and the collective to explore and find optimal solutions related to resource consumption and allocation without heavily sacrificing individual freedoms. The new paradigm is applicable in centralized, decentralized, and hybrid design approaches, where AI participates in different levels of decision-making, in the form of a recommender tool or a simulation model.

In Chapter 2, an AI control method (NODEC) is introduced. NODEC shows high performance when addressing high dimensional complex dynamics, which are often found in several sustainability oriented applications. Other control baselines are also considered, such as deep reinforcement learning, which often achieve lower performance in terms of energy and accuracy and require considerable training time. NODEC shows high versatility and its potential applications have a wide range, from control of pandemics to power grid stabilization. The implicit energy regularization properties and the ability of NODEC can be used to achieve potential value-sensitive design, especially for participatory centralized controls.

Chapter 3 identifies and quantifies sustainability concepts related to consumption. As a decentralized approach it avoids several shortcomings of centralized approaches, especially in applications where individual freedom and values are involved. This

chapter introduces a valuable framework for building and maintaining knowledge bases that combine expert data, personal preferences, and product characteristics to calculate sustainability ratings. Most importantly, real-world applicability is tested on two European retailers. Causal impact analysis on transaction data shows an increase of expenditure towards more sustainable products for participants that used and trusted the system. Most participants stated that their awareness of sustainable consumption increased after using the proposed recommender system. Chapter 3 also showcased the underlying complexity of designing such recommender systems. Collective values and outcomes are difficult to estimate in a privacy-preserving manner, when following the design applied during the study. Participant retention was challenging during the field study, indicating that participants may not be willing to evaluate each product at each market visit at the cost of their own comfort. These insights were very useful for designing the methods of Chapter 4.

Chapter 4 introduces the personalized sustainable basket problem, extending the application of Chapter 3 to baskets. This problem optimizes different objectives related to environmental impact, nutritional information, and taste. Several multi-objective optimization algorithms are evaluated. Each algorithm calculates recommended consumer baskets, which respect the consumer personal preferences while reducing environmental footprint. A novel gradient guided genetic algorithm (G3A) is also proposed to solve the problem, and achieves similar or higher performance across several objectives when compared with the baselines. All algorithms operate in a fully decentralized manner. Collective information is used in the hybrid version of G3A (Hybrid-G3A), where aggregates over individual data are calculated in a centralized system in privacy-preserving manner following Chapter 5. The hybrid model achieves higher performance in environmental impact objective without creating a considerable compromise of personal objectives compared to G3A.

Chapter 5 addressed the privacy-preserving challenges observed in the fields of study of Chapter 3. For this, Chapter 5 introduced privacy-preserving calculations of aggregate values, such as total consumption and average preferences, by introducing a novel differential privacy framework. In this framework, individuals are free to choose their privacy level, while organizations and institutions can decide the upper bound of privacy to ensure the usability and accuracy of the aggregated data. The application of the framework to electric consumer data shows efficient calculations of aggregate values with little loss of privacy for individuals. The outcomes of this chapter were extremely helpful for including collective goals and constraints in hybrid design approaches of Chapter 4.

Chapter 6 complements the models discussed in Chapter 2 by introducing AI Pontryagin, a time-dependent version of NODEC. In this chapter, time-dependent controls are evaluated on different settings, where AI Pontryagin is compared against optimal control methods. AI Pontyagin achieves similar performance to optimal control methods, requiring less analytical effort and seemingly computation time. The implicit energy regularization properties of neural ODE controllers are further confirmed, while controls can be calculated even when the selection of driver nodes is non-optimal. These outcomes offer useful insights about the desirable value-sensitive

properties of minimum impact neural ODE controls for centralized socio-technical systems.

In conclusion, this thesis provides several analytical, applied, and theoretical outcomes that support the design and implementation of value-sensitive AI for sustainable decision-making. These decisions may be very hard to tackle by humans alone. Although, proposed applications of AI methods and techniques may change in the future, developing the paradigm of designing value-sensitive AI is the key outcome for this thesis. Sustainable decision-making will always be a relevant problem for humanity, especially as technology progresses. Therefore, having another form of intelligence that respects our values and supports our decisions will be reassuring for our future.

Appendices

Appendix

A

Centralized Control on Critical Infrastructures with Neural Ordinary Differential Equation Control

Data Availability Statement

The experiment data that support the findings of this study are openly available in the NODEC IEEE Dataport repository [186] at `https://dx.doi.org/10.21227/gdqj-am79`. The code that fully reproduces the above experiments is found as a code ocean capsule [185] at `https://codeocean.com/capsule/1934600/tree`. A fully functioning code library [184] written in python for NN control with NODEC is found at `https://github.com/asikist/nnc` with coding examples and more applications.

A.1 Kuramoto Oscillators

Curriculum Learning

A curriculum learning procedure is used to train Kuramoto models. The algorithm is described below in Algorithm A.1.

Synchronization Loss Before Convergence

In this section, we describe one of the results presented in Figure 2.3b in more detail. We observe that NODEC takes more time to converge to a synchronized state in the example illustrated in Figure A.1. We also observe that NODEC requires a higher amount of control energy before reaching the synchronized state Figure 2.3a. Once synchronicity is reached, the NN can adapt and produce lower energy controls. This might not be the case for feedback control, which has a constant term ζ multiplied by the driver matrix values.

Algorithm A.1: Curriculum training process of NODEC. A procedure that gradually increases total time is introduced in this algorithm. Here we present a stochastic procedure, but a deterministic procedure is also possible.

Result: $\boldsymbol{w}$
Init:: $\boldsymbol{x}_0$, $\boldsymbol{w}$, $\boldsymbol{f}(\cdot)$, ODESolve$(\cdot)$, Optimizer$(\cdot)$, $J(\cdot)$, $\boldsymbol{x}^*$;
Params:: η, epochs, stepSize ;
epoch $\leftarrow 0$;
$T \leftarrow 0$;
while epoch $<$ epochs **do**
  $t \leftarrow 0$;
  $c \sim \mathcal{U}(0, 1)$;
  $T \leftarrow T + 2 \cdot c$;
  $\boldsymbol{x} \sim \eta_N(0_1^N, 1_1^N)$;
  meanLoss $\leftarrow$ **List**;
  minLoss $\leftarrow \infty$;
  while $t < T$ **do**
    $\boldsymbol{x}(t)$, hasNumInstability $\leftarrow$ ODESolve$(\boldsymbol{x}, t, t +$ stepSize$, \boldsymbol{f}, \hat{\boldsymbol{u}}(\boldsymbol{x}; \boldsymbol{w}))$;
    if Not hasNumInstability **then**
      meanLoss $\leftarrow$ (stepSize$/T) \cdot J(\boldsymbol{x}(t), \boldsymbol{x}^*)$);
      if minLoss $> J(\boldsymbol{x}_t, \boldsymbol{x}^*)$ **then**
        minLoss $\leftarrow J(\boldsymbol{x}_t, \boldsymbol{x}^*)$;
      end
    end
    $t \leftarrow t +$ stepSize ;
  end
  Optimizer.update($\boldsymbol{w}$, meanLoss + minLoss);
end

A.2 SIR-Type

NN Architecture

Here, we provide some technical details and an overview of the GNN architecture presented in Figure 2.5 and complements the code. The final output of a NN is a control vector $\hat{\boldsymbol{u}}(\boldsymbol{X}(t))$. The input of the GNN is a tensor $\Psi \in \mathbb{R}^{4 \times N \times \hat{d}}$, where $\hat{d}$ is the maximum degree of the graph. An element $\Psi_{k,j,i}$ of the tensor represents the k-th state of the j-th neighbor of node i. The j-th neighbor of node i is fixed via any permutation of neighbors prior to training. The operation that constructs a tensor Ψ from the input state matrix $\boldsymbol{X}(t)$ is referred to as "neighborhood embedding". GNN applies an operation for each node that aggregates the state values over all neighboring nodes and produces a hidden state tensor $\boldsymbol{H}(\boldsymbol{\Psi})$. This hidden state is provided to the consecutive layers, and a hidden state matrix (or embedding)

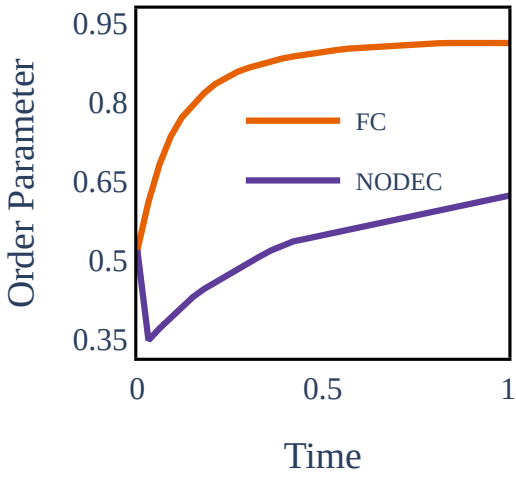


FIGURE A.1: Early order parameter values based on Figure 2.3b.

$\boldsymbol{Z} \in \mathbb{R}^{4\times N}$ is calculated, with same dimensions as the input state matrix $\boldsymbol{X}$. This matrix $\boldsymbol{Z}$ is provided again as input to the GNN structure described above (see left side of Figure 2.5) and a new tensor Ψ is calculated based on the neighborhood embedding procedure. Providing the calculated hidden state matrix $\boldsymbol{Z}$ as an input to the GNN is termed "message passing" [335], and is a typical procedure when training GNNs. Message passing essentially allows the NN to calculate a hidden state representation for each node i but also leverage information of non-adjacent neighbors for the calculation after the first repetition. We observe that allowing the message passing process to repeat 4 times maximizes the performance of the network for the current control task. For example, in the second iteration of the above procedure, the input tensor $\boldsymbol{\Psi}$ of the GNN contains a representation calculated by a functional on an aggregation over all the state i of all adjacent nodes j' of each neighbor j of node i, based on the NN parameters. In conclusion, the GNN architecture aims to learn a state representation $\boldsymbol{Z}$ that can be used to produce efficient control signals that take into account the states of non-neighboring nodes of each driver. After the last message propagation is finished, the mean over the channels is calculated over the hidden state matrix $\langle \boldsymbol{Z} \rangle_0$ generating a hidden state vector $\boldsymbol{z} \in \mathbb{R}^N$.

Reinforcement Learning

In this section, we focus on the technical details of the RL baseline we used in the main paper. Reinforcement learning is often described as "model-free" and addresses the (i) prediction problem and (ii) control problem [145]. We note that RL approaches may suffer from credit assignment challenges, where a reward signal is

uninformative regarding the specific actions (especially in terms of time) that help reach the goal [180]. In contrast to RL, the proposed NODEC is not model-free and the underlying gradient descent is directly calculated from the loss function. Therefore, we do not need to consider value prediction and credit assignment. It is possible to design a model-free NODEC by learning the underlying system dynamics simultaneously with control, which could be an interesting future extension of our work. Note that a direct performance comparison between RL and NODEC in terms of target loss may be considered unfair especially towards RL methods, unless extensive hyperparameter optimization is performed beforehand.

We first implement SIR-type dynamics as an RL environment. The softmax activation function and budget assignment discussed in Section 2.4 take place in the environment and RL computes the softmax logit values over all nodes. Reinforcement learning is allowed to interact with the environment in a fixed interaction interval $\Delta t = 10^{-2}$, similar to NODEC. A2C and SAC implementations are taken from StableBaselines3[1]. Both implementations were tested for different parameter sets and trained for at least 50000 steps. Unfortunately, no implementation was able to "flatten the curve" considerably better than random control. Next, we use the TD3 implementation from Tianshu[2], which currently showcases high-speed benchmarks and allows more customization of policy/critic architectures. The corresponding RL training takes around 17 seconds per epoch, whereas NODEC takes approximately 5.5 seconds per epoch. Neither TD3 or NODEC fully utilized the GPU in terms of computing and memory resources, often staying below 50% of usage, while memory utilization usually was below 10GB per method.

We show an overview of the hyperparameters that we use to train TD3 in Table A.1. For more detailed explanations of these hyperparameters, see Ref. [155] and the Tianshu documentation[3]. Several baseline architectures in RL frameworks are often fully-connected multilayer perceptrons. Still, we observe that the GNN presented in Figure 2.5 was more efficient in converging rewards in less computation time. We trained all models for 100 epochs and stored and evaluated the best model. In SAC and A2C, one training environment was used, whereas TD3 was sampling from two independent environments simultaneously due to its computational speed.

In terms of parameters both the TD3 policy network and NODEC GNN have exactly the same learning parameters (weights), but training is very different, as the gradient flows described in Figure 2.1 and Algorithms 2.1 and 2.2 cannot happen. The value function is now used for the calculation of similar gradients by predicting the cumulative reward signal. We studied several possible reward designs, and in the end we rigorously tested the following rewards:

The first reward signal we tested is calculated based on the mean number of infected nodes belonging to the target sub-graph $\bar{I}_{G^*}(t)$ at time t:

$$\rho_1(t) = -(\bar{I}_{G^*}(t))^2 \Delta t. \tag{A.1}$$

[1] `https://github.com/DLR-RM/stable-baselines3`

[2] `https://github.com/thu-ml/tianshou`

[3] `https://tianshou.readthedocs.io/en/latest/api/tianshou.policy.html?highlight=td3#tianshou.policy.TD3Policy`

Although this reward seemingly provides direct feedback for an action, it also leads to several challenges. First, it does not necessarily flatten the curve, but it minimizes the overall infection through time. Such a reward could, for instance, potentially reinforce actions that lead to "steep" peaks instead of a flattened infection curve, as in practice it minimizes the area under the $I(t)$ curve. Furthermore, as current containment controls may have effect if applied consistently and in the long term, such reward design suffers from temporal credit assignment, since the reward value depends on a long and varying sequence of actions. Finally, any actions that happen after the peak infection occurrence will still be rewarded negatively, although such actions do not contribute to the goal minimization.

The next reward

$$\rho_2(t) = \begin{cases} 0 & , \text{if } t < T \\ -(\max_{t \leq T} \bar{I}_{G^*}(t))^2 & , \text{otherwise} \end{cases} \tag{A.2}$$

is designed to overcome the aforementioned shortcomings. This reward signal is sparse through time, as it is non-zero only at the last step of the control when the infection peak is known. The main property of interest of Relation (A.2) is that it has the same value as the loss that we used to train NODEC (see Relation (2.22). This reward signal also suffers from credit assignment problems. As the reward is assigned at a fixed time and not as a direct result of the actions that caused it, the corresponding reward dynamics is non-Markovian [336]. To address challenges caused by rewards with non-Markovian properties, reward shaping [337] and recurrent value estimators [338] can be used. Furthermore, n–step methods or eligibility traces can be evaluated if we expect the reward signal to be Markovian but with long and/or varying time dependencies.

The final reward $\rho_3(t)$ that we evaluated and used in the presented results is designed with two principles in mind:

$$\sum_t \rho_3(t) \propto \max_{t \leq T} (\bar{I}_{G^*}(t))^2 \tag{A.3a}$$

$$\arg\min_{t \leq T} \sum_t \rho_3(t) = \arg\max_{t \leq T} (\bar{I}_{G^*}(t)). \tag{A.3b}$$

Following those principles, the reward signal is approximately proportional to and provides information about the value of the infection peak used in the NODEC loss calculation. The reward sum minimizes exactly at the time when peak infection occurs. This property is expected to reduce effects of temporal credit assignment. When aiming to replace the proportionality in Relation (A.3a) with an equality, we reach the following reward signal design presented in the main thesis chapter Relation (2.23):

$$\rho_3(t) = \begin{cases} 0, & \text{if } \bar{I}_{G^*}(t) \leq \max_{\tau < t}(\bar{I}_{G^*}(\tau)) \\ -\bar{I}^2_{G^*}(t) + (\max_{\tau < t} \bar{I}_{G^*}(\tau))^2, & \text{otherwise,} \end{cases} \tag{A.4}$$

which is equivalent to the reward function of Relation (2.23) in the main paper. It is straightforward to show that Relation (2.23) indeed satisfies

$$\sum_t \rho_3(t) = \max_{t \leq T} (\bar{I}_{G^*}(t))^2 \tag{A.5}$$

and Relation (A.3b). This reward greatly improved performance without resorting to recurrent value estimators or further reward shaping. Still, after all proposed reward design and hyper-parameter optimization, NODEC has a higher performance (see Figure A.2), although TD3 performs better than random control.

In Figures A.3a and 2.7 the dynamic controls of both RL and NODEC seem to focus on protecting the target sub-graph by containing the infection as it spreads. In contrast to targeted constant control, they succeed in doing so by protecting driver nodes outside the target sub-graph. When comparing the dynamic control patterns, the budget allocation of NODEC seems to be much more concentrated on specific nodes, and it creates more often contiguous areas of containment.

In Figure A.2, we also show the evolution of the proportion of susceptible $S(t)$, infected $I(t)$, recovered $R(t)$, and contained individuals $Y(t)$. We observe that TCC and NODEC show clear signs of flattening the curve by preserving the highest susceptibility fraction and lowest recovery fraction at time T, which can be interpreted as less susceptible nodes becoming infected and needing to recover. The random method outperforms the other frameworks in terms of effective containment fractions, as random control assignments at each time step let the disease spread such that higher proportions of infected individuals $I(t)$ are reached in the target sub-graph and therefore drivers with high infection fractions are effectively contained when controlled. Although low energy effective containment might seem favorable at first sight, it is not optimal in terms of flattening the curve with restricted budget, as it allows high infection fractions to occur within an area of interest. Budget restrictions often do not allow to fully constrain the spread in all infected nodes.

In Figure A.4, we observe that although RL does not converge in terms of critic and actor loss, it still converges to a higher reward. This confirms that RL is capable of controlling continuous dynamics with arbitrary targets, but it requires significant parameterization and training effort to have good stable value estimates.

Finally, we tried to examine transfer learning capabilities from NODEC to RL. A closer look at Figure 2.5 reveals that the parameterized graph neural architecture used for NODEC and RL can be the same, i.e. there are no weights in the decision network layers of Figure 2.5. This means that the architectures trained with NODEC can be used as the "logit" action policy in RL, showcasing an effective use of transfer learning. In the given example, the RL policy network starting with trained NODEC parameters, is further trained for 100 episodes. After training, RL had a similar performance as NODEC since both methods flatten the curve at approximately $\bar{I}_{G^*} = 0.0788$. This means that RL did not improve the solution generated by NODEC. This example can be used to illustrate the interplay between NODEC and RL and how they can be used in synergy, e.g. when back-propagating through continuous dynamics is too expensive for high number of epochs. Reinforcement learning can be used as a meta-heuristic on top of NODEC, and the latter can be treated as an alternative to imitation learning.

Table A.1: Tested and evaluated hyperparameters for the TD3 reinforcement learning baseline.

Hyper-Parameter	Value	Tested Values
Actor learning rate	0.0003	0.0003, 0.003, 0.03
Actor architecture	GNN	GNN, FC
Critics learning rate	0.0001	0.0001, 0.001, 0.01
Critics architecture	FC	FC
τ (Polyak update parameter)	0.005	0.005, 0.05
γ (discount factor)	0.99	0.5, 0.8, 0.99, 1
exploration gaussian noise mean	0.01	0, 0.01. 0.1
update frequency of actor parameters	4 epochs	1–4 epochs
policy noise	0.001	0.001. 0.01, 0.1
noise clip	0.5	0.5, 0.2
reward normalization	True	True, False

A.3 Other Notes

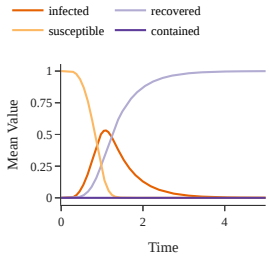
Hardware and code

Our experiments were mainly conducted on a dedicated server that was equipped with a NVIDIA TITAN RTX GPU, 64GB of RAM, and an Intel I9 9900KF 8-core processor. Partial code tests with assertions were conducted to examine (i) stiffness, (ii) numerical errors or bugs, and (iii) validity and similarity of the same dynamics controlled by different models. For the majority of the experiments seeds are fixed and initial states parameters are stored in data files to enable reproducibility. ODEsolve and sample experiments may be affected by stochasticity on different machines. Based on statistical testing, we observe that with a good initialization and NN hyperparameter optimization, NODEC performs close to the reported values. Future works under provided repository, may perform extensive hyper-parameter studies dedicated to specific dynamics and graphs. The average training time of NODEC per task is between 5-10 minutes depending on the complexity of the task. Baseline methods calculations and parameterizations would also take minutes, making time performance comparable.

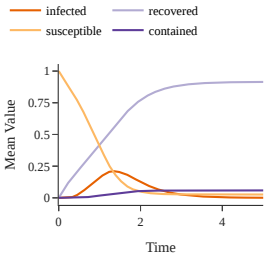
The project code can be found on GitHub `https://github.com/asikist/nnc` under MIT license. Numerical experiments are stored in the experiment folder (please check github `readme` for more details).

ODE Solvers and Stiffness

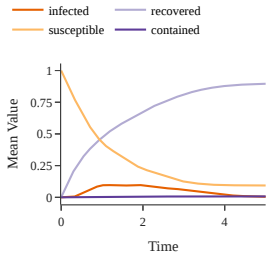
We used the Dormand–Prince solver [158] for the majority of our numerical experiments (in particular for training). For evaluating our results, we use a specific



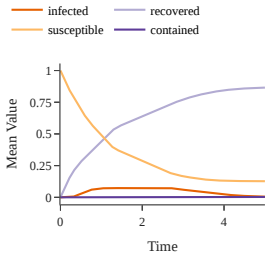
(a) No control (F) baseline.



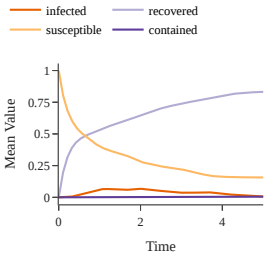
(b) Random control (RND) baseline.



(c) Reinforcement learning control (RL) baseline.



(d) NN control (NODEC) baseline.

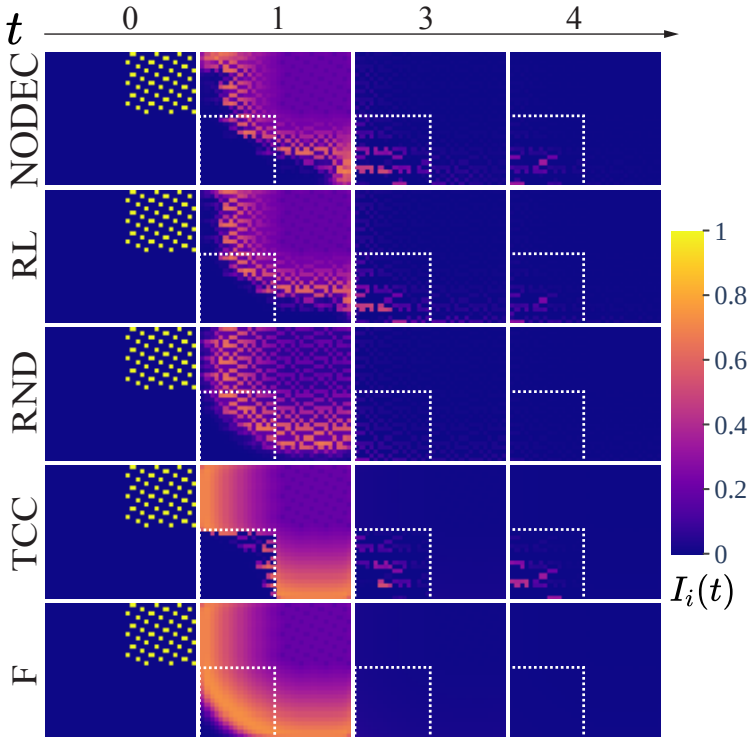


(e) Targeted constant control (TCC) baseline.

FIGURE A.2: Evolution of the proportion of susceptible, infected, recovered, and contained individuals for all baselines in the target subgraph G^*.

method, which allows the controller to change the control signal at constant time intervals. This choice allows us to compare control errors and energy costs without considering interaction frequency bias that occurs when one method outperforms another method because the solver allowed it to interact more often with the system and produce more tailored control signals. Adaptive step length helps the NN to learn controls for variable interaction intervals and approximate continuous control better. We performed small-scale unit tests with VODE [158] against Dormand–Prince, Runge–Kutta, and implicit Adams implementations, and we noticed that for most systems numerical errors were negligible.

The goal of Chapter 2 is to evaluate the ability of NODEC to learn controls within a solver. In future works that aim at controlling large-scale systems, different ODE solvers may be chosen according to the system's stiffness and performance requirements of the application. Whenever dynamics and training had high VRAM requirements, the adjoint method was used, mainly the implementations from Refs. [136, 339].

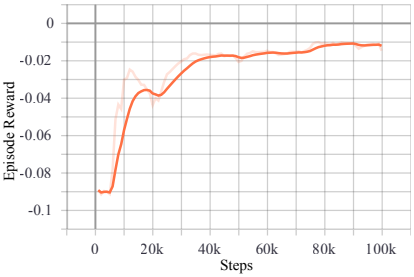


(a) Infection spread on lattice for all baselines.

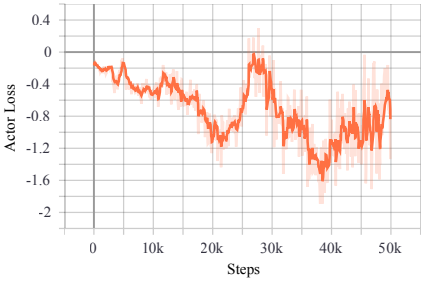
FIGURE A.3: Spread of infection for baselines.

Adaptive Learning Rate Training

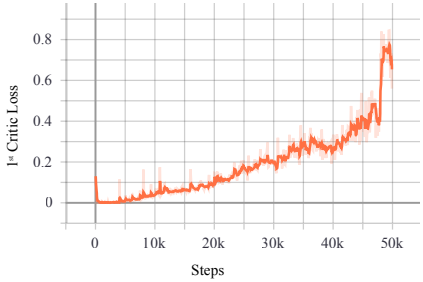
Learning rate plays an important role on reaching a low energy control. In order to determine the optimal learning rate values we propose the adaptive learning rate scheme found in the Algorithm A.2.



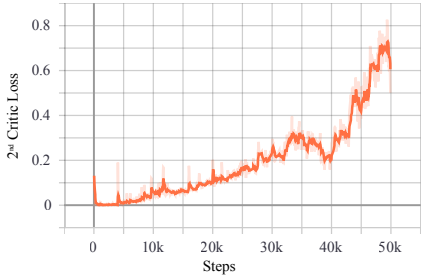
(a) Total episode reward as TD3 trains.



(b) Actor loss as TD3 trains.



(c) First critic loss.



(d) Second critic loss.

Figure A.4: RL learning performance evaluation plots using Tensorboard with 0.8 smoothing.

Algorithm A.2: Adaptive Learning rate training process of NODEC.

Result: $\boldsymbol{w}$
Init:: $\boldsymbol{x}_0$, $\boldsymbol{w}$, $(f)(\cdot)$, ODESolve$(\cdot)$, Optimizer$(\cdot)$, $J(\cdot)$, $\boldsymbol{x}^*$;
Params:: η, epochs, q, tolRatio;
epoch $\leftarrow 0$;
bestLoss $\leftarrow \infty$;
bestParams $\leftarrow$ copy$(\boldsymbol{w})$;
previousLoss;
while epoch $<$ epochs **do**
    $t \leftarrow 0$;
    $\boldsymbol{x} \leftarrow \boldsymbol{x_0}$;
    $\mathrm{X}_{t_0}^{T}$, hasNumInstability $\leftarrow$ ODESolve$(\boldsymbol{x}, 0, T, \boldsymbol{f}, \hat{\boldsymbol{u}}(\boldsymbol{x}(t); \boldsymbol{w}))$;
    if $J(\mathrm{X}_{t_0}^{T}, \boldsymbol{x}^*) >$ tolRatio $\cdot$ previousLoss$\vee$ hasNumInstability **then**
        $\boldsymbol{w} \leftarrow$ bestParams;
        $\eta \leftarrow \eta q$;
        Optimizer.reset();
        Optimizer.learningRate $\leftarrow \eta$;
    end
    else
        if $J(\boldsymbol{x}, \boldsymbol{x}^*) <$ bestLoss **then**
            bestParams $\leftarrow$ copy$(\boldsymbol{w})$;
            bestLoss $\leftarrow J(\boldsymbol{x}, \boldsymbol{x}^*)$;
        end
        previousLoss $\leftarrow J(\boldsymbol{x}, \boldsymbol{x}^*)$;
        Optimizer.update$(\boldsymbol{w}, J(\boldsymbol{x}, \boldsymbol{x}^*))$;
    end
end

A.4 Nomenclature

The notation used in the respective chapter is summarized in Tables A.3, A.5 and A.7.

Table A.3: Nomenclature Part I for Section 2.3.

t_0	The initial time for control of a dynamical process. Often we may also use $t = 0$ without loss of generality.
T	The terminal time for control of a dynamical system.
Δt	A time-interval.
$G(\mathrm{V}, \mathrm{E})$	A graph represented as an ordered pair of a set of nodes V and a set of edges E.
N	The number of nodes in a graph $N = \lvert\mathrm{V}\rvert$.
$\boldsymbol{A}$	The adjacency matrix that represents a graph G. It has non zero elements $\boldsymbol{A}_{i,j} \neq 0$ if and only if nodes i, j are connected.
$\boldsymbol{x}(t)$	A vector $\boldsymbol{x}(t) \in \mathbb{R}^N$, which denotes the state of a dynamical system at time t.
$\boldsymbol{x}^*$	A vector that denotes the target state of a dynamical system.
$\dot{\boldsymbol{x}}(t)$	Newton's dot notation for differentiation of the system state.
M	The number of driver nodes, i.e. nodes that can be controlled in a graph. As the driver nodes is a subset of all the nodes we have $M \leq N$.
$\boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}(t)))$	The system evolution function that denotes the dynamic interactions between nodes and drivers when calculating the state derivative.
$\boldsymbol{u}(\boldsymbol{x}(t))$	A feedback control signal function $\boldsymbol{u}(\boldsymbol{x}(t)) \colon \mathbb{R}^N \to \mathbb{R}^M$ calculated based on the system state at time t.
$\boldsymbol{B}$	A driver matrix $\boldsymbol{B} \in \mathbb{R}^{N \times M}$, where $\boldsymbol{B}_{i,m} = 1$ if node i is the m-th driver node and receives a control signal $u_m(t)$.
$E\left(\boldsymbol{u}(\boldsymbol{x}(t))\right)$	The total energy value of a control signal calculated from time t_0 until time t.
$\hat{\boldsymbol{u}}(\boldsymbol{x}(t))$	A control signal value calculated by NODEC.
$\boldsymbol{w}$	Vector with NN parameters for NODEC.
$\mathrm{X}_{t_0}^T$	The state trajectory between t_0 and T. An ordered set of state vectors $\boldsymbol{x}(t), t \in [t_0, T]$.
$J(\mathrm{X}_{t_0}^T, \boldsymbol{x}^*; \boldsymbol{w})$	Learning and control objective function for NODEC. In the current work, we evaluate control goals $\boldsymbol{x}^*$ that are achieved over a state trajectory $\mathrm{X}_{t_0}^T$.
$\Delta \boldsymbol{w}$	Gradient descent update for NN parameters.
η	Learning rate hyper-parameter for gradient descent.
$\boldsymbol{h}(t)$	The hidden state evolution function used in the neural ODE paper [136].
$\mathrm{ODESolve}(\boldsymbol{x}(t), t, T, \boldsymbol{f}, \boldsymbol{u})$	The function that denotes a numerical ODE solving scheme.

Table A.5: Nomenclature Part II (Coupled Oscillators) for Section 2.4.

ω_i	Natural frequency of oscillator (node) i.
K	Coupling constant.
$\hbar(x_i - x_j)$	2π-periodic function function that couples oscillators. Often the sine function is used, s.t. $h(\cdot) = \sin(\cdot)$.
$\boldsymbol{x}^{\diamond}$	Synchronized steady state of coupled oscillator system.
$\boldsymbol{D}$	the graph degree diagonal matrix $\boldsymbol{D}$ of G, where all off-diagonal elements are 0 and diagonal elements are equal to the degree $\boldsymbol{D}_{i,i} = d_i$ of the corresponding node.
$\boldsymbol{L}$	the graph Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ of G.
$\boldsymbol{L}^{\dagger}$	Pseudo-inverse of the graph Laplacian matrix $\boldsymbol{L}$ of G.
$\boldsymbol{b}^{(\mathrm{FC})}$	The feedback control gain vector of the FC baseline.
$r(t)$	Order parameter, which denotes the synchronization of coupled oscillators.
ζ	Scaling parameter for feedback control baseline.
τ	Timestep.
Ξ	Number of timesteps for discretizing the time period $[0, T]$.
ξ	Timestep index used to calculate discretized approximations of continuous time metrics.
$r_{\mathrm{NODEC}(t)}$	Order parameter value achieved under NODEC control at time t.
$E_{\mathrm{NODEC}(t)}$	Total energy value achieved under NODEC control at time t.
$r_{\mathrm{FC}(t)}$	Order parameter value achieved under feedback control baseline at time t.
$E_{\mathrm{FC}(t)}$	Total energy value achieved under feedback control baseline control at time t.

Table A.7: Nomenclature Part III (Disease Spreading) for Section 2.4.

$S_i(t)$	Susceptible fraction of individuals at node i at time t.
$I_i(t)$	Infected fraction of individuals at node i at time t.
$R_i(t)$	Recovered fraction of individuals at node i at time t.
$Y_i(t)$	Contained fraction of individuals at node i at time t.
$\boldsymbol{X}(t)$	The matrix representation of the state, where the state vectors $\boldsymbol{s}, \boldsymbol{i}, \boldsymbol{r}, \boldsymbol{s}$ as rows. Columns represent the node index in the graph.
G^*	Target sub-graph, i.e. the subset of nodes that we are interested to reduce the peak infection.
β	Infection rate.
γ	Recovery rate.
c_j	A number sampled from a uniform distribution $c_j \sim \mathcal{U}(0, 1)$ to calculate random control.
$\mathfrak{b}$	Control budget. A linear constraint on maximum total control that can be applied on the graph at time t.
$\rho(t)$	Reward signal for reinforcement learning techniques.
d_i	The degree of a node i.
$\hat{d}$	The maximum degree.
Ψ	Input tensor for convolutional NN of the GNN.
$\boldsymbol{Z}$	Output of hidden layers to be used for message propagation in the GNN.

Appendix

B

Value-Sensitive Design for Decentralized Sustainable Product Ratings

B.1 Supplementary Figures

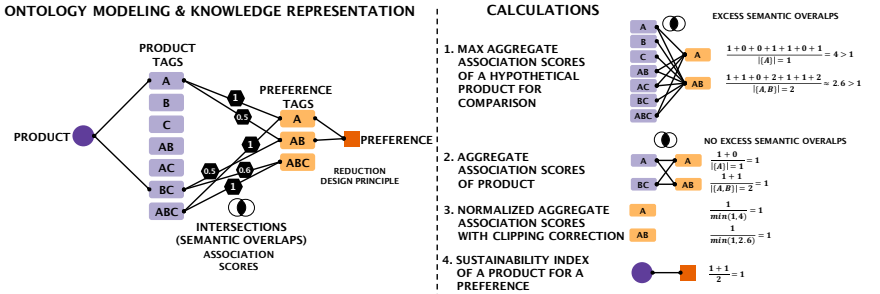


Figure B.1: An alternative calculation of sustainability index illustrated in Figure 5. This calculation uses a theoretical normalized aggregate association score that does not respect the reduction design principle of Section B.3. To avoid violation of the bounding properties of Relation (B.19), the clipping methodology introduced in Section B.3 is used.

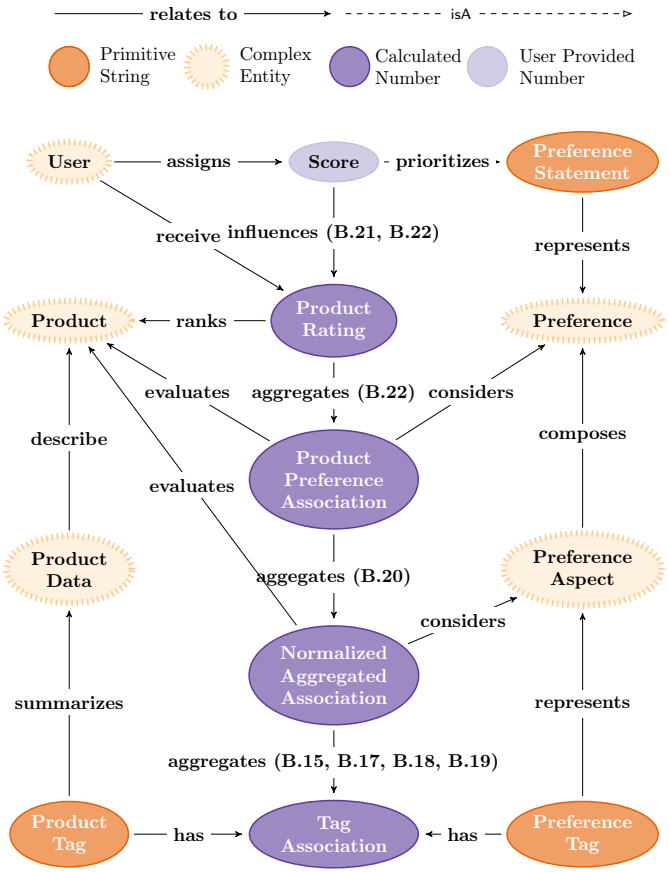
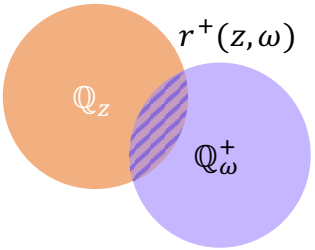
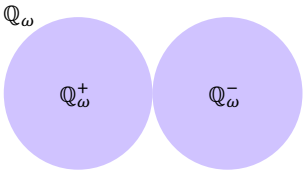


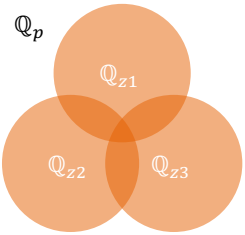
Figure B.2: Product rating is calculated by utilizing the ontology. There are ontology relations that contain the relevant equation from the current chapter and Chapter 3 in parenthesis.



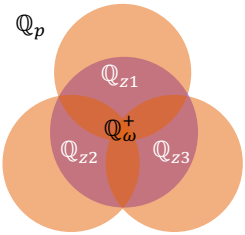
(a) Overlapping product tag and preference tag concepts.



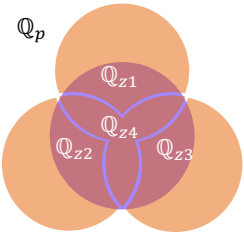
(b) The semantic space that defines a preference tag is the union of the concepts, which support or oppose its existence.



(c) A product concept as a union of its product tag concepts.

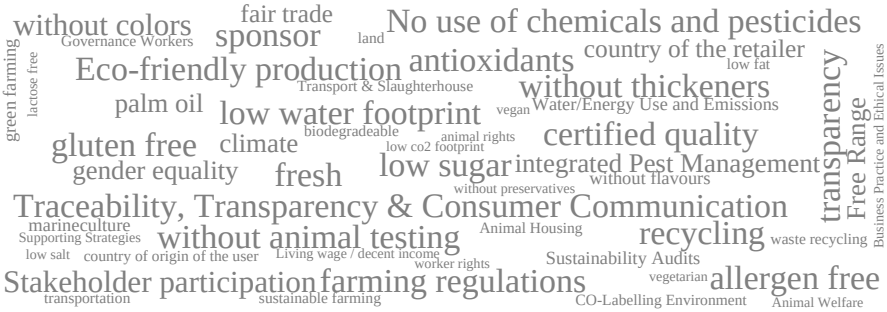


(d) Semantic overlap between a product and a preference tag.



(e) Introduction of new product tag according to the reduction design principle.

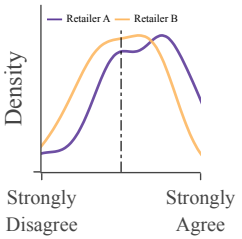
FIGURE B.3: Venn diagrams that illustrate the operations between sets of preference and product tags primitive concepts.



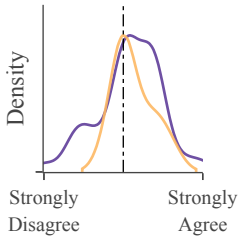
(a) Preference tags.

(b) Product tags.

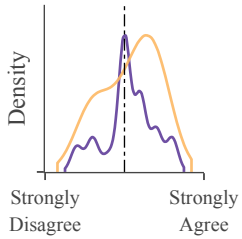
FIGURE B.4: Wordclouds of preference and product tags. The size is proportional to the number of products assigned to each tag.



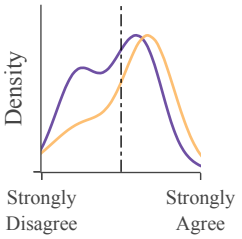
(a) B.I.7: Finding additional product information in the app was easy.



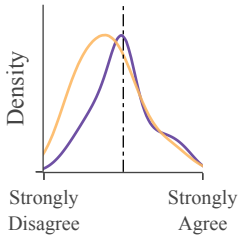
(b) B.I.8: The product info justify the rating of the products.



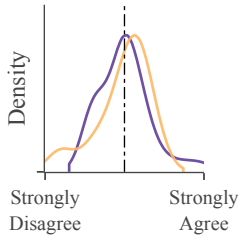
(c) B.II.6: The products with high rating match my preferences.



(d) B.II.13: I discovered new products while using the rating functionality.



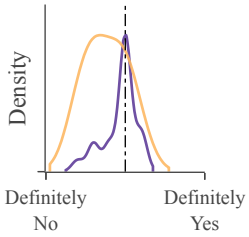
(e) B.II.14: The app does not take into account my preferences when calculating the ratings.



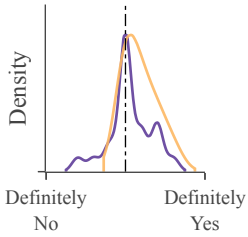
(f) B.III.1: The app could capture my actual preferences.

Figure B.5: Questions regarding the application that indicate acceptance from users.

(a) B.IV.3: The products I used to buy before using the app achieve high rating in the app.



(b) B.IV.4: The products I used to buy before using the app achieve low rating.



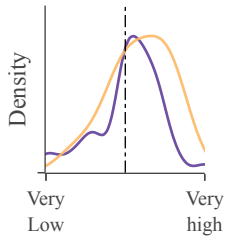
(c) B.IV.5: The products I will buy in the future are the ones with high rating in the app.

(d) B.IV.6: The products I will buy in the future are the ones with low rating in the app.

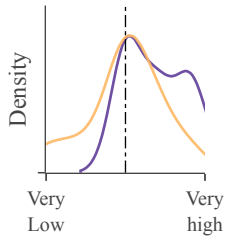
(e) B.IV.9: I am more aware about sustainability aspects after using the app.

Figure B.6: Questions regarding sustainability awareness and future purchases.

(a) The rating of the app.



(b) The product information I read in the app.



(c) Their price.

Figure B.7: The products I finally chose during this study are because of:

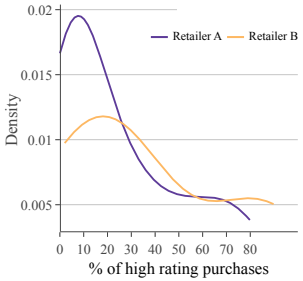
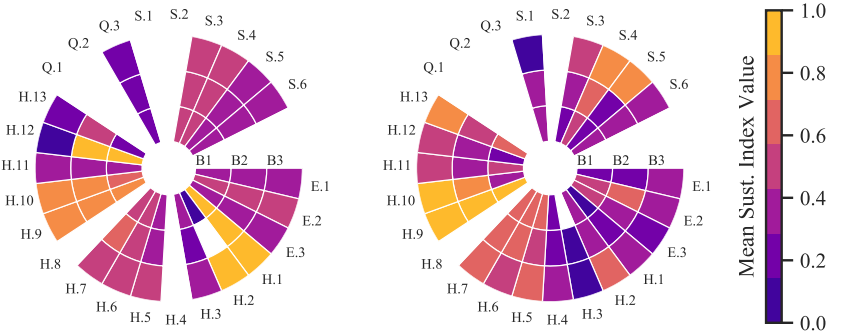


FIGURE B.8: Estimate the % of highly rated products you finally bought:



(a) Retailer A. (b) Retailer B.

FIGURE B.9: Evaluation of mean normalized sustainability index (colorscale) per category across the different price bins per category as denoted in Table B.24 for all products of each retailer.

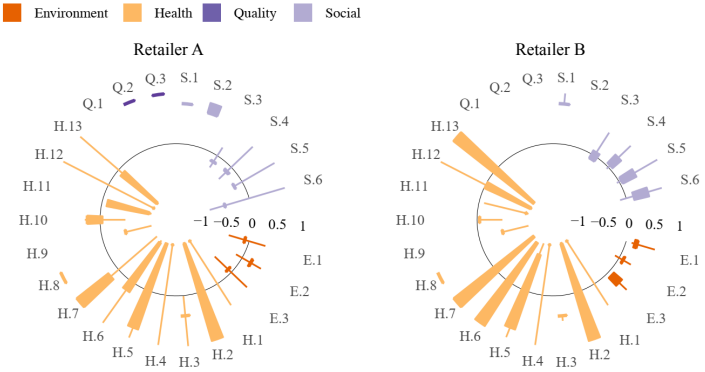
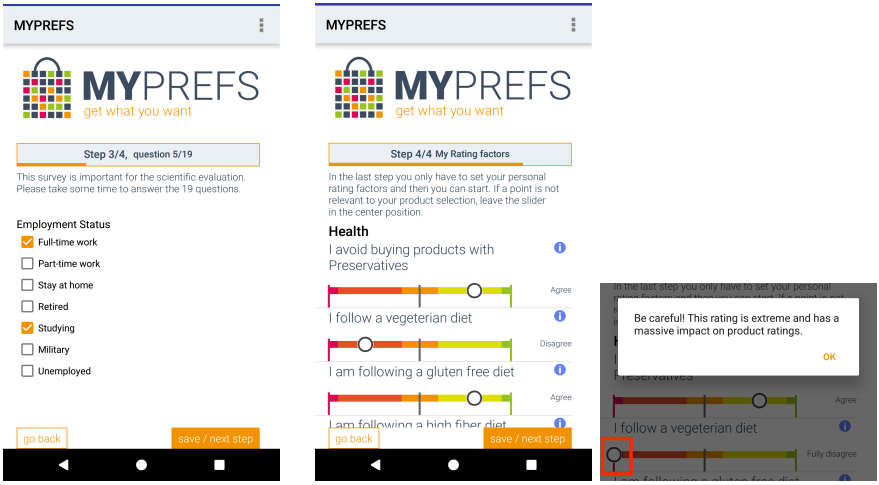


Figure B.10: The distribution of sustainability index per preference for all products in both retailers (Retaieler A left, Retailer B right).

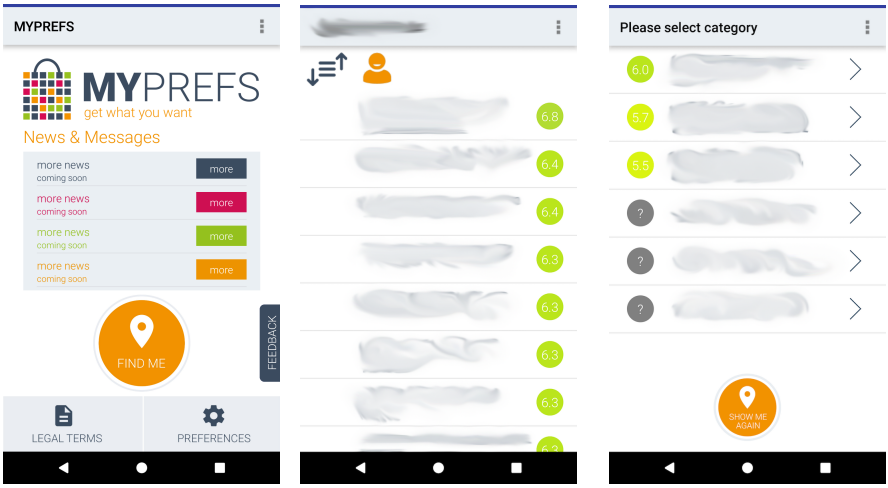


(a) Answering the demographics questions in the entry survey.

(b) Assigning preference scores.

(c) Notification for extreme preference scores, which are also explained to the user in the app tutorial.

Figure B.11: Survey and the preference user interfaces of ASSET.

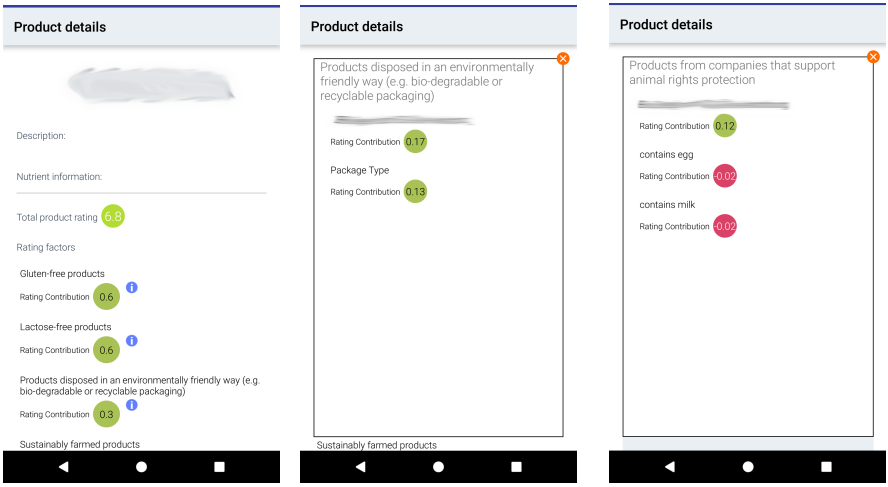


(a) The main screen of the application.

(b) Presenting nearby product categories.

(c) Receiving product ratings.

FIGURE B.12: Using the application.



(a) The amount a preference contributes to the product rating.

(b) Product tags that contribute a positive amount to the rating.

(c) Product tags that contribute a negative amount to the rating.

FIGURE B.13: Preference and product tag contribution to the rating. This feature improves the explainability of the product rating.

B.2 Supplementary Tables

Table B.1: **E**nvironmental Preferences. The preferences setting options provided in value-sensitive smartphone application.

Preference	Description	**ID**
1. Products disposed in an environmentally friendly way (e.g. biodegradable or recyclable packaging)	Products and product packages that can be recycled efficiently or are biodegradable and do not cause environmental hazard when disposed of.	E.1
2. Products produced and distributed in an environmentally friendly way	A product has a sustainable lifecycle if during manufacturing, distribution, consumption and disposal: it has low CO2 and water footprint, low transportation costs, it can be efficiently recycled and/or biodegradable and produces no waste that is harmful to the environment. Some examples include toxic waste, high CO2 emissions during production, usage of palm oil during production that leads to deforestation etc.	E.2
3. Sustainably farmed products	Sustainable farming is achieved by a company by applying farming methods such as Integrated Pest Management, No-till agriculture, biodynamic and permaculture. For sea products, aquaculture is preferred instead of free fishing. The countries where the company produces its products should have clear and fair regulations regarding farming and land-management.	E.3

TABLE B.3: **Q**uality Preferences. The preferences setting options provided in value-sensitive smartphone application.

Preference	Description	**ID**
1. Award winning or high quality certified products.	There are several officially recognized organizations that offer awards and/or certifications to companies that produce high-quality products.	Q.1
2. Fresh products	Fresh products are directly brought from production to the shelves of the supermarket. Fresh products usually tend to be richer in nutrients. Also it is less likely for fresh products to cause health problems to the digestive system.	Q.2
3. Locally originated and domestic products.	I prefer products that the country of origin is the same as the one I am living in.	Q.3

Table B.5: Social Preferences. The preferences setting options provided in value-sensitive smartphone application.

Preference	Description	**ID**
1. Products evaluated with auditing processes that rely on sustainability criteria.	All the processes that enhance and enable sustainability should be checked and validated from 3rd parties. Auditing processes based on sustainability criteria support companies in adopting more sustainable processes and also increase their transparency.	S.1
2. Products from companies that actively contribute to public and social good	I prefer supporting companies, that sponsor charities, scholarships, R&D, social activities	S.2
3. Products from companies that support animal rights protection	Brands that respect and support animal rights should avoid factory farming, animal testing and any kind of animal abuse/mistreatment during the production of their products.	S.3
4. Products from companies that support fairness and equality in the workplace	Brands that treat their workers equally and respect their rights. Such brands should take action against any discrimination between their employees and promote gender and race equality.	S.4
5. Products from companies with transparent activities	I prefer buying products from companies that are open to sharing the impact and nature of their operation.	S.5
6. Products from fair trade label companies	Fair trade focuses on human rights in general. Companies should respect human rights and trade with producers in developing and developed countries in a fair manner, offering fair prices for raw materials and not abusing local laws for fair resource sharing, like public water sources, fair land management etc.	S.6

TABLE B.7: Health Preferences. The first 7 preferences setting options provided in value-sensitive smartphone application.

Preference	Description	**ID**
1. Allergen-free products	Food allergens are typically naturally-occurring proteins in foods or derivatives of them that cause abnormal immune responses. Eight ingredients cause about 90% of food allergy reactions: Milk (mostly regarding children), Eggs, Peanuts, Tree nuts, like walnuts, almonds, pine nuts, brazil nuts, and pecans, Soy, Wheat and other grains with gluten, including barley, rye, and oats, Fish (mostly in adults), Shellfish (mostly in adults)	H.1
2. Gluten-free products	A gluten-free diet is a diet that strictly excludes gluten, a mixture of proteins found in wheat and related grains, including barley, rye, oat, and all their species and hybrids (such as spelt, kamut, and triticale). The inclusion of oats in a gluten-free diet remains controversial, and may depend on the oat cultivation and the frequent cross-contamination with other gluten-containing cereals.	H.2
3. High-protein products	Protein is one of the three macronutrients, along with carbs and fat. It is beneficial for building muscle. Protein serves a number of important functions in your body. It is made up of individual amino-acids, including many that your body cannot create on its own. Protein is the main component of your muscles, bones, skin and hair. These tissues are continuously repaired and replaced with new protein.	H.3
4. Lactose-free products	A lactose free diet means eating foods that have no lactose. Lactose is a sugar that is a normal part of dairy products. Some people do not break down lactose well. They may not have enough lactase, the enzyme that breaks lactose down in the body. Or, their body may create lactase variants that do not work properly.	H.4
5. Low fat products	Fats are a type of nutrient that you get from your diet. It's a major source of energy. It helps you absorb some vitamins and minerals. Fat is needed to build cell membranes, the vital exterior of each cell, and the sheaths surrounding nerves. It is essential for blood clotting, muscle movement, and inflammation. It is essential to eat some fats, though it is also harmful to eat too many. For long-term health, some fats are better than others. Good fats include monounsaturated and polyunsaturated fats. Bad ones include industrial-made trans fats. Saturated fats fall somewhere in the middle. Saturated fats raise your LDL (bad) cholesterol level. High LDL cholesterol puts you at risk for heart attack, stroke, and other major health problems. Trans fats can raise LDL cholesterol levels in your blood. They can also lower a person's HDL (good) cholesterol levels.	H.5
6. Low salt products	There is also some evidence that too much salt can damage the heart, aorta, and kidneys without increasing blood pressure, and that it may be harmful for bones, too.	H.6

Table B.9: Health Preferences. The remaining preferences setting options provided in value-sensitive smartphone application.

Preference	Description	**ID**
7. Low sugar products	Added sugar is known to cause heart diseases. Sugar delivers "empty calories" — calories unaccompanied by fiber, vitamins, minerals, and other nutrients. Too much added sugar can replace healthier foods from a person's diet.	H.7
8. Products rich in antioxidants	Antioxidants come up frequently in discussions about good health and preventing diseases. Their nature is to prohibit (and in some cases even prevent), the oxidation of other molecules in the body. Oxidation is a chemical reaction that can produce free radicals, leading to chain reactions that may damage cells. The term "antioxidant" is mainly used for two different groups of substances: industrial chemicals which are added to products to prevent oxidation, and natural chemicals found in foods and body tissue which are said to have beneficial health effects. It is often debated whether they actually prevent diseases, which antioxidant(s) are needed from the diet and in what amounts beyond typical dietary intake.	H.8
9. Products without artificial colours or flavor enhancers.	Artificial colors and flavoring enhancers are used to produce coloring effects and improve food taste. In general these substances are exhaustively tested in labs before they are used in food production. Still, it is not fully determined whether they cause health problems in the long term.	H.9
10. Products without preservatives	A preservative is a substance or a chemical that is added to products such as food, beverages, pharmaceutical drugs, paints, biological samples, cosmetics, wood, and many other products to prevent decomposition by microbial growth or by undesirable chemical changes. Preservatives are used to prolong the shelf-life of the product but may cause health problems in the long term.	H.10
11. Products without thickeners, stabilizers or emulsifiers	Emulsifiers allow water and oils to remain mixed together in an emulsion, as in mayonnaise, ice cream, and homogenised milk. Stabilizers, thickeners and gelling agents, like agar or pectin (used in jam for example) give foods a firmer texture. While they are not true emulsifiers, they help to stabilize emulsions. These additives may cause health problems in the long term.	H.11

Table B.11: Health Preferences. The remaining preferences setting options provided in value-sensitive smartphone application.

Preference	Description	**ID**
12. Vegan products	Vegans choose not to consume dairy, eggs or any other products of animal origin, in addition to not eating meat like the vegetarians. Veganism was originally defined as "the principle of emancipation of animals from exploitation by man."	H.12
13. Vegetarian products	The Vegetarian Society defines a vegetarian as follows: "A vegetarian is someone who lives on a diet of grains, pulses, legumes, nuts, seeds, vegetables, fruits, fungi, algae, yeast and/or some other non-animal-based foods (e.g. salt) with, or without, dairy products, honey and/or eggs. A vegetarian does not eat foods that consist of, or have been produced with the aid of products consisting of or created from, any part of the body of a living or dead animal. This includes meat, poultry, fish, shellfish*, insects, by-products of slaughter** or any food made with processing aids created from these."	H.13

Table B.13: An example of association score values and their corresponding meaning. RDA refering to "Recommended Daily Allowance" [340]. LDL refering to "Low-density lipoprotein".

association score	product tag $\boldsymbol{z}$	preference tag $\boldsymbol{\omega}$	**Meaning**
$r(z,\omega) = 1$	vegetable	vegetarian diet	p fully supports c via z
$r(z,\omega) \in (0,1)$	10% RDA Vitamin C	healthy diet	p partially supports c via z
$r(z,\omega) = 0$	contains sugar	animal rights	z is irrelevant to ω
$r(z,\omega) \in (-1,0)$	contains LDL	healthy diet	ω partially opposes z
$r(z,\omega) = -1$	contains eggs	vegan diet	ω fully opposes z

TABLE B.14: **Alternative approaches.** Overview of online sources that provide sustainability ratings. Most of these approaches limit their scope to evaluation of brands rather than a broad spectrum of products. They are online web approaches with limited integration to shopping processes in retailer shops. Explainability is not explicitly provided and data may only be collected in a centralized fashion. They do not always capture a broad spectrum of sustainability goals. No rigorous evaluation with field studies has been shown how they impact sustainable consumption.

Organization	Sector Scope	Target	Computation	Main Focus	Rating Type
rankabrand [341]	General	Brand	Crowdsourced	Environment, Social and Health	Ordinal
GoodGuide® [342]	Personal Care, Household Supplies, Babies and Kids products	Product	Ingredient analysis based on official regulations	Environment and Health	Continuous
Shop Ethical! [343]	General	Brand	Screening and reports for praises and/or criticism towards companies based on expert weighting	Social and Environmental	Ordinal
THE GOOD SHOPPING GUIDE [344]	General	Companies, Brands	Qualitative and quantitative evaluation based on expert criteria.	Environment, Animals, People, Ethical and Other Factors	Ordinal
The Green Stars Project [345]	General	Companies, Brands, Services, Products etc.	User based reviews and rating assignment	Social and Environmental	Ordinal
WikiRate [346]	General	Company	Allows for implementation of any rating based on the data the website offers	Environmental, Social and Governmental Concerns	Any
Behind the Brands [347]	Food	Brand	Expert based	Social, Environmental and Ethical	Continuous, ordinal

Table B.16: **Data Sources**. The data sources used for the construction of the used ontology. Websites accessed on January 2020.

Source Type	Institute	Website
Retailer Database	Coop Estonia	coop.ee
	Winkler Markt	www.winklermarkt.at
Online Datastores	GS1	gs1.org
	ecoinvent	ecoinvent.org
Experts	AINIA	ainia.es
	BIA	bia.ee
	ETH	ethz.ch
	Ethical Consumer Workshop 2017	ethicalconsumer.org
	LCM	lcm.at
	VKI	vki.at
Crowdsourcing	Wikipedia	wikipedia.org
	Social Impact Data Hack	sidh2017.ut.ee

Table B.18: Total number of participants per surveu per retailer.

Retailer	Survey	# Users
Retailer A	Entry	323
	Exit	44
	Preferences	315
Winkler Markt	Entry	69
	Exit	12
	Preferences	66

TABLE B.20: **Causal Impact Analysis**. Best kNN matches for treatment group selected on the DTW euclidean distance on high mean rating products weekly expenditure. The control group is selected by using covariates for matching between users. Since different values of k result in different control group per combination of criteria the best value of k is selected via a dynamic time warping of weekly expenditure before treatment. The warping window size is 1, meaning that the algorithm searches 1 week before or after to match the expenditure of the current weekly expenditure between control and treatment.

Retailer	Monthly Total Expenditure	Distribution of Monthly Expenditure per Category	Monthly Representations	Distance	k
retailer A	-	-	✓	0.030852	5
	-	✓	-	0.037871	5
	-	✓	✓	0.032995	2
	✓	-	-	0.032736	4
	✓	-	✓	0.033386	5
	✓	✓	-	0.033722	2
	✓	✓	✓	0.035229	4
retailer B	-	✓	-	0.056523	1
	-	✓	✓	0.061028	2
	✓	-	-	0.066083	2
	✓	-	✓	0.055792	2
	✓	✓	-	0.061923	1
	✓	✓	✓	0.059597	2

Table B.22: The summary of the causal impact analysis for Retailer A and Retailer A users. The reported decimal values in the table are rounded.

Retailer		**Average**			**Cumulative**			p
		value	s.d.	95% c.i.	value	s.d.	95% c.i.	
A	Actual	0.8	-	-	14.7	-	-	$\leq$ 0.001
	Prediction	0.6	0.0	[0.5, 0.6]	10.8	0.3	[10.1, 11.4]	
	Absolute Effect	0.2	0.0	[0.2, 0.2]	4.0	0.3	[3.3, 4.6]	
	Relative Effect	36.7%	3.1%	[30.7%, 42.8%]	-	-	-	
B	Actual	0.6	-	-	10.1	-	-	$\leq$ 0.001
	Prediction	0.4	0.0	[0.3, 0.5]	7.1	0.8	[5.4, 8.7]	
	Absolute Effect	0.2	0.0	[0.1, 0.3]	2.9	0.8	[1.4, 4.7]	
	Relative Effect	41.0%	11.8%	[19.2%, 65.4%]	-	-	-	

Table B.24: The price bins used for classifying a product according to its price per unit in Figure B.9. Each product price is compared against prices of other products in the same category. Package specific quantities are not taken into account.

Label	Normalized Price Range	Description
B1	$(0, 0.33)$	The 33% of lowest product prices in category
B2	$(0.33, 0.66)$	Between 33% and 66% product prices
B3	$(0.66, 1)$	The 66% and 100% of highest product prices in category

B.3 Supplementary Methods

Ontology

In the used product ontology, product characteristics are summarized in the form of words or phrases, the **product tags**. Each product tag z can be assigned to one or more products and it summarizes concepts and characteristics of a product p. Every product p can be assigned to multiple product tags. The set of all product tags denotes the semantic space of product characteristics. Product tags are generated based on the data available from the data sources. For example $p =$ "cabbage" is associated with the product tag $z =$ "vegetable".

The consumer preferences (see Table B.1-B.7) are also represented in the consumer preference ontology. Several challenges arise when defining an universal golden standard regarding a sustainable consumption behavior. Therefore, a personalized view on sustainable consumption is evaluated per consumer. Each preference is presented to the consumer in form of a statement c. Each statement is accompanied by a description that explains the sustainability concepts composing this preference, e.g. $c =$ "I prefer vegan products.". A consumer u then assigns numerical values to each preference statement c, to express support, opposition or neutrality towards the preference statement. This numerical value is referred to as preference score $s_{u,c}$. The preference score is bound in the range of $[0, 2 \cdot \overline{s}]$, where $\overline{s}$ is the mean value of the range. The minimum value of the score implies that a consumer fully opposes a sustainability preference. The maximum value implies fully supports a sustainability preferences. The mean value of the scale implies no consumer preference regarding a sustainability criterion. An assigned preference score is stored in the device of

Table B.26: (Max-Min) Normalized price per category and normalized sustainability index for retailer A. Relevant to Figure B.9a.

Preference	Correlation	p-value
E.1	0.03	0.21
E.2	-0.08	0.00
$^{+}$E.3	0.10	0.00
H.1	0.09	0.70
H.2	1.00	-
H.3	-0.05	0.47
H.4	-	-
H.5	-0.02	0.21
H.6	-0.03	0.17
H.7	-0.04	0.12
H.8	-	-
$^{-}$H.9	-0.10	0.01
H.10	0.14	0.23
H.11	0.03	0.74
$^{-}$H.12	-0.88	0.02
H.13	0.13	0.11
Q.1	-	-
Q.2	-	-
$^{+}$Q.3	0.12	0.01
S.1	-	-
S.2	-	-
S.3	0.00	0.81
S.4	-0.05	0.00
S.5	0.02	0.21
S.6	0.07	0.00

a consumer u, and represents personalization. The rating system calculates the product rating $\varrho(p, u)$ for a given product p and a consumer u based on the preference

scores. The introduced method extends the semantic differential methodology for evaluating associations of product characteristics and sustainability preferences [348, 349].

Preference statements usually express *abstract* complex sustainability aspects. These aspects are decomposed into simpler ones creating a hierarchical ontology. A word or phrase, referred to as preference tag ω, represents an aspect of a preference statement. For example, c = "I prefer products that can be disposed in a sustainable manner." is composed by two sustainability preference tags: ω_1 = "biodegradability", which is the ability of the product to dissolve within an acceptable time and without harming the environment [350], and ω_2 = "recycling capability", which denotes whether a product can be recycled in an efficient and environmentally friendly way [351]. In other words, a preference tag $\omega \in \Omega_c$, which belongs to a preference tag set Ω_c of a preference c, compresses and represents information regarding the concepts that compose the preference statement c.

The main challenge of designing an ontology for a sustainable consumption ontology is to define quantifiable associations between the semantic spaces of product tags and preference tags. To enable numerical calculation for product ratings we use the ontological design as sketched in Figure B.2. Several ontological connections are introduced to calculate the aggregate support or opposition of a product to a consumer's preferences.

The ontology of sustainability concepts has hierarchical structures of concepts [352], where one concept is composed of several other concepts $q \in \mathbb{Q}$. If it is not feasible to further decompose a concept, then such a concept is referred to as a *primitive concept*[1]. A **semantic association framework** introduces a logic for quantifiable semantic associations between a product tag z and a preference tag ω. Examples of such associations can be found on Table B.13. The associations between tags are quantified in a shared semantic space $\mathbb{Q}$, which contains all concepts relevant to products and sustainability preferences. The space is defined with the following two assumptions:

1. The individual elements of this space are *primitive concepts* that cannot be decomposed into other concepts within the defined sustainability scope.
2. Complex concepts are represented as sets that contain all the *primitive concepts* that compose them. For instance, the preference tag ω = "vegan" may be decomposed to the set $\mathbb{Q}_\omega$ = { "no animals involved in production", "no animal products involved in production", ...} where $\mathbb{Q}_\omega \subseteq \mathbb{Q}$.

Suppose a product tag expresses one or more *primitive concepts*. The set of those concepts is defined as $\mathbb{Q}_z \subseteq \mathbb{Q}$. Following the same logic, let a preference that its existence is denoted by the union of *primitive concepts* such as $\mathbb{Q}_\omega^+$. An association between the product and the preference tag is defined as the overlap between the *primitive concepts* that each tag represents $\mathbb{Q}_z \cap \mathbb{Q}_\omega^+$ as shown in Figure B.3a. This overlap is maximized when all *primitive concepts* that compose the preference tag

[1]The definition of primitive concepts serves more as a theoretical tool, with which the scope of the sustainability can be determined.

also compose the product tag. The following positive association score is defined:

$$r^{+}(z,\omega) = \frac{\left|\mathbb{Q}_z \cap \mathbb{Q}_\omega^{+}\right|}{\left|\mathbb{Q}_\omega^{+}\right|} \tag{B.1}$$

The score is bounded, since:

$$\mathbb{Q}_z \cap \mathbb{Q}_\omega^{+} \subseteq \mathbb{Q}_\omega^{+} \Leftrightarrow r^{+}(z,\omega) \in [0,1] \tag{B.2}$$

For example, let a product tag and the relevant *primitive concepts*: z = "vegetable" with $\mathbb{Q}_z$ = { "minimal CO_2 footprint", "plant part", "not animal product"}. Suppose the following two preference tags, which are decomposed to *primitive concepts*, ω_1 = "vegetarian diet" with $\mathbb{Q}_{\omega 1}$ = { "plant part", "not animal product"} and ω_2 = "sustainable production" $\mathbb{Q}_{\omega 2}$ = { "minimal CO_2 footprint", "minimal water footprint", "no toxic waste"}. The "vegetable" product tag is composed of concepts which are enough to guarantee a "vegetarian diet". Yet, the "vegetable" product tag is composed of some but not all of the concepts that guarantee a sustainable production.

A product tag may oppose the existence of a preference tag concept. For example the concept of "animal product" in food, fully opposes a "vegan diet" preference. When a product contains animal products, then it is definitely not vegan. Partial opposition is also possible, as for example the concept "contains lactose" indicates that a product probably contains animal products, as lactose is a protein mainly found in milk [353]. Therefore, the need to define opposite associations arises. The negative *primitive concepts* of a preference tag are defined as a set $\mathbb{Q}_\omega^{-}$. For that, an association is defined when the concept in a product tag is mutually exclusive with concepts defining a preference tag. In this case a negative association score evaluates the overlap of concepts belonging to the set of simple concepts of the product tag $\mathbb{Q}_z \cap \mathbb{Q}_\omega^{-}$. The negative association score is defined as:

$$r^{-}(z,\omega) = \frac{\left|\mathbb{Q}_z \cap \mathbb{Q}_\omega^{-}\right|}{\left|\mathbb{Q}_\omega^{-}\right|} \tag{B.3}$$

The negative score is also bounded, since:

$$\mathbb{Q}_z \cap \mathbb{Q}_\omega^{-} \subseteq \mathbb{Q}_\omega^{-} \Leftrightarrow r^{-}(z,\omega) \in [0,1] \tag{B.4}$$

The preference tag is defined by the union of its negative and positive concepts $\mathbb{Q}_\omega = \mathbb{Q}_\omega^{+} \cup \mathbb{Q}_\omega^{-}$ as shown in Figure B.3b. E.g. the preference tag ω = "vegan" can be decomposed to the supporting *primitive concepts* $\mathbb{Q}_\omega^{+}$ = "plant part".

Product tags that contain *primitive concepts* that both oppose and support a preference tag are also possible. In such case, the sum of positive and negative scores is calculated. For this calculation, it is assumed that positive simple concepts cancel out negative concepts and vice versa. The association score between a product tag and a preference tag is defined as:

$$r(z,\omega) = r^{+}(z,\omega) - r^{-}(z,\omega) \overset{B.2,B.4,B.5}{\Leftrightarrow} r(z,\omega) \in [-1,1] \tag{B.5}$$

This is denoted as the **individual bounding property** of the score and applies to all association scores between any product and preference tags.

A product is often related to a set of product tags $\mathbb{z}_p$, therefore it can be represented by the union of all *primitive concepts* of these tags:

$$\mathbb{Q}_p = \bigcup_{z \in \mathbb{z}_p} \mathbb{Q}_z \tag{B.6}$$

An example of how a product p is decomposed to the *primitive concepts* of the product tags it consists of is found in Figure B.3c.

In practice, the calculation of the association score is performed by knowledge systems that rely on (i) expert knowledge, (ii) crowdsourcing and (iii) machine learning. As illustrated on line 4 in Table B.13, the products that contain high quantities of LDL, oppose the preference tag "healthy diet", e.g. the health of the cardiovascular system [354]. This is quantified via a negative association score value in the range $(-1, 0)$. The negative threshold value is assigned to the "contains eggs" product tag, which fully opposes the "vegan" preference tag, as shown on row 5 of Table B.13.

Associations between products p and preference tags are calculated by an aggregated association score $\eta(p, \omega)$. According to Relation (B.6), the intersection between all *primitive concepts* related to a product and a preference tag is used for such calculation:

$$\eta(p, \omega) = \frac{\left|\mathbb{Q}_p \cap \mathbb{Q}_\omega^+\right|}{\left|\mathbb{Q}_\omega^+\right|} - \frac{\left|\mathbb{Q}_p \cap \mathbb{Q}_\omega^-\right|}{\left|\mathbb{Q}_\omega^-\right|} \tag{B.7}$$

Note that, it is challenging to assign *primitive concepts* directly to products. A fixed sustainability scope supports the identification of overlaps and the existence of primitive concepts. The definition of primitive concepts is challenging when the ontology is under construction and the sustainability scope is not fixed. Conceptual overlaps are common in real-world scenarios where product tags are usually derived from labels or certifications, which are related to several *primitive concepts*, e.g.the fair trade label. Intersection between a product tag and a preference tag $r(z, \omega)$ quantifies their shared semantic space, consisting of at least one primitive concept. The individual intersections between product tags and preference tags $r(z, \omega)$ can be used for the approximation of the aggregated association $\eta(p, \omega)$. The number of available product tags is considerably lower than the number of products worldwide. When the product tags share *primitive concepts*, it may prove challenging to calculate this intersection, especially when the *primitive concepts* are not identified. Shared product tag *primitive concepts* introduce overlaps between intersections of different product tags and a preference tag. Such overlaps introduce an error ϵ in the approximation of the aggregate associations $\eta(p, \omega)$. Reducing the overlaps between product tags of the same product, minimizes this error, as shown in *Lemma* 2.

Lemma 2. *The aggregated association $\eta(p, \omega)$ between a product p and a preference tag ω is approximated with error ϵ by the sum of tag associations $r(z, \omega)$ of each related product tag $z \in \mathbb{z}_p$ with the preference tag ω, assuming that the primitive concept overlaps between product tags are minimized $\bigcap_{z \in \mathbb{z}_p} \mathbb{Q}_z \to 0$, such that $\epsilon \to 0$.*

Proof. A product is defined as the union of associated *primitive concepts* that product tags represent. Therefore the aggregated association for $z \in \mathbb{z}_p$ is:

$$\begin{aligned}
\eta(p,\omega) &= \frac{\left|\mathbb{Q}_p \cap \mathbb{Q}_\omega^+\right|}{\left|\mathbb{Q}_\omega^+\right|} - \frac{\left|\mathbb{Q}_p \cap \mathbb{Q}_\omega^-\right|}{\left|\mathbb{Q}_\omega^-\right|} \overset{Relation\ (B.6)}{\Leftrightarrow} \\
\eta(p,\omega) &= \frac{\left|\left(\bigcup_z \mathbb{Q}_z\right) \cap \mathbb{Q}_\omega^+\right|}{\left|\mathbb{Q}_\omega^+\right|} - \frac{\left|\left(\bigcup_z \mathbb{Q}_z\right) \cap \mathbb{Q}_\omega^-\right|}{\left|\mathbb{Q}_\omega^-\right|} \Leftrightarrow \\
\eta(p,\omega) &= \frac{\left|\bigcup_z \left(\mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right)\right|}{\left|\mathbb{Q}_\omega^+\right|} - \frac{\left|\bigcup_z \left(\mathbb{Q}_z \cap \mathbb{Q}_\omega^-\right)\right|}{\left|\mathbb{Q}_\omega^-\right|}
\end{aligned} \tag{B.8}$$

Each nominator can be further analyzed using the general form of the Inclusion-Exclusion principle [355]. The first fraction nominator is expanded as, for any subset of product tags $\emptyset \neq \mathbb{z}'_p \subseteq \mathbb{z}_p$, and then all intersection between a preference tag and more than one product are isolated to determine overlaps:

$$\begin{aligned}
\left|\bigcup_z \left(\mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right)\right| &= \sum_{\emptyset \neq \mathbb{z}'_p \subseteq \mathbb{z}_p} \left[(-1)^{|\mathbb{z}'_p|-1} \left|\bigcap_{z \in \mathbb{z}'_p} \mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right|\right] \\
&= \sum_{z \in \mathbb{z}_p} \left|\mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right| \\
&\quad + \underbrace{\sum_{\substack{|\mathbb{z}'_p|>1 \\ \mathbb{z}'_p \subseteq \mathbb{z}_p}} \left[(-1)^{|\mathbb{z}'_p|-1} \left|\bigcap_{z \in \mathbb{z}'_p} \mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right|\right]}_{\epsilon^+}
\end{aligned} \tag{B.9}$$

Thus:

$$\left|\bigcup_z \left(\mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right)\right| = \sum_{z \in \mathbb{z}_p} \left|\mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right| + \epsilon^+ \tag{B.10}$$

Respectively it is shown that:

$$\left|\bigcup_z \left(\mathbb{Q}_z \cap \mathbb{Q}_\omega^-\right)\right| = \sum_{z \in \mathbb{z}_p} \left|\mathbb{Q}_z \cap \mathbb{Q}_\omega^-\right| + \epsilon^- \tag{B.11}$$

The terms ϵ^+ and ϵ^- are the overlap correction terms introduced by the Inclusion-Exclusion principle. These terms express the semantic overlap between different product tags of a product and a single preference tag. Therefore, it is possible now to expand *Relation* B.8:

$$\begin{aligned}
\eta(p,\omega) &= \frac{\sum_z \left|\mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right| + \epsilon^+}{\left|\mathbb{Q}_\omega^+\right|} - \frac{\sum_z \left|\mathbb{Q}_z \cap \mathbb{Q}_\omega^-\right| + \epsilon^-}{\left|\mathbb{Q}_\omega^-\right|} \overset{Eq.\ B.1\ \&\ B.3}{\Leftrightarrow} \\
\eta(p,\omega) &= \sum_z r^+(z,\omega) - \sum_z r^-(z,\omega) + \epsilon \overset{Eq.\ B.5}{\Leftrightarrow} \\
\eta(p,\omega) &= \sum_z r(z,\omega) + \epsilon
\end{aligned} \tag{B.12}$$

where:

$$\epsilon = \frac{\epsilon^+}{\left|\mathbb{Q}_\omega^+\right|} + \frac{\epsilon^-}{\left|\mathbb{Q}_\omega^-\right|} \tag{B.13}$$

Since the intersection between sets is a commutative operation, it can be derived from Relation (B.9) that:

$$\left|\bigcap_{z\in\mathbb{z}'_p} \mathbb{Q}_z \cap \mathbb{Q}_\omega^+\right| = \left|\mathbb{Q}_\omega^+ \cap \left(\bigcap_{z\in\mathbb{z}'_p} \mathbb{Q}_z\right)\right| \tag{B.14}$$

Equations B.9, B.10 and B.11 show that if the overlaps $\bigcap_{z\in\mathbb{z}_p} \mathbb{Q}_z \to 0$ between product tags are minimized then $\epsilon^+, \epsilon^- \to 0 \Rightarrow \epsilon \to 0$. Thus the lemma is proved. □

For example assume the product $p =$ "orange-lettuce-rice salad" in Figure B.3c, which is associated with the product tags $\mathbb{z}_p =$ { "vegetable", "fruit", "cereal"}. The product tags $z_1 =$ "cereal", $z_2 =$ "fruit" and $z_3 =$"vegetable" share several *primitive concepts*, such as "plant part". Each product tag has a positive association score with the preference tag $\omega =$ "vegetarian". As it is showcased in Figure B.3d there are several overlaps between associations, due to the shared primitive concepts of the product tags. Summing all association scores with the preference tag "vegetarian" introduces errors because of the shared "primitive concepts". The calculation of error correction terms ϵ^+, ϵ^- requires all possible combinations of intersections of product tags with shared *primitive concepts*. Such calculation in the worst case requires an exponential time complexity of $O(2^n)$ for every aggregate association score. Therefore, the aggregation of association scores may become infeasible for an ontology with a high number of shared *primitive tags* amongst associations per product. This challenge can be addressed in the construction of the ontology, by indentifying and isolating overlapping *primitive concepts*. In the previous example, this can be achieved by creating a new product tag $z_4 =$ "plant part" and assigning it to all products that have the "vegetable" or "fruit" tag. All the association scores between the "vegetable" and "fruit" product tags are now reduced by an amount δ, such that $r(z, \omega) = r(z, \omega) - \delta$ with $z \in \{z_1, z_2\}$. All associated preference tags with the product tags "fruit" and "vegetable" can now be associated to the product tag "plant part" with association score $r(z_4, \omega) = \delta$, as illustrated in Figure B.3e. In such case, all terms of Relation (B.13) are equal to 0, since all possible intersection are equal to the empty set.

Based on the above example to avoid overlaps, a generic **reduction design principle** is proposed during the ontology design:

1. If a product tag contains all the *primitive concepts* of another product tag, then only one is chosen and assigned to a product.
2. If there are overlaps of *primitive concepts* between two product tags of the same product, but neither can be omitted because their non-shared *primitive concepts* are important, then the intersection of their *primitive concepts* should be treated as a separate product tag and assigned to all products these tags are associated with. The shared *primitive concepts* are omitted from the original product tags.

The introduction of the reduction design principle minimizes overlap error $\epsilon \to 0$ and therefore the aggregate association can now be calculated as:

$$\eta(p, \omega) = \sum_{z} r(z, \omega) \tag{B.15}$$

This calculation has linear complexity $O(n)$ to the number of product tag concepts in each aggregate association calculation. Yet, the reduction design principle methodology introduces a quadratic $O(n^2)$ complexity in regards to product tags, once during the creation of the ontology, as each tag needs to be compared against each other to determine overlaps. The reduction design principle is illustrated in Figure B.3e.

If the aggregated association is 0, then it is not possible to determine whether a product supports or opposes a preference tag, since it contains equally enough positive and negative concepts. The uncertainty is treated as a product having no information. Analyzing and comparing the aggregate associations between different preferences can be used for the identification of possible *trade-offs* and *rebound effects*. If no overlapping occurs ($\epsilon = 0$) between the associations, then the individual bounding property is also extended to the aggregated association. The **aggregated association bounding property** is the following:

$$\begin{gathered} \frac{\left|\mathbb{Q}_p \cap \mathbb{Q}_\omega^+\right|}{\left|\mathbb{Q}_\omega^+\right|} - \frac{\left|\mathbb{Q}_p \cap \mathbb{Q}_\omega^-\right|}{\left|\mathbb{Q}_\omega^-\right|} \in [-1, 1] \overset{Relation\ (B.12), \epsilon=0}{\Leftrightarrow} \\ \eta(p, \omega) \in [-1, 1] \end{gathered} \tag{B.16}$$

Application of the reduction design principle reduces overlaps, yet it may increase the amount of tags and associations in the ontology, when tags are decomposed to disjoint tags with less primitive concepts. Furthermore, successful application relies on the ability of the expert or system to identify and break down associations between preference and product tags. Therefore a trade-off is introduced between efficiency, performance and maintenance of the ontology.

Product Rating Mechanism

The ontology is used to calculate a distributed and privacy-preserving product rating value $\varrho(p, u) \in \mathbb{R}$ between a product p and a user u. This is achieved by implementing a product rating system with rating design principles of the content based recommender systems [356, 357]. The product rating is designed to use the aggregated association scores $\eta(p, \omega)$.

Comparable Aggregated Associations

Different products may satisfy the same preference tag via product tags that are related to different *primitive concepts*. For that reason, comparison of aggregated

associations for the same preference tag and different products are not easy to interpret, i.e. each product may have completely unique characteristics that satisfy each preference. To be able to compare how different products satisfy a preference, a reference product can be used to normalize all aggregate association scores. The reference products are defined as follows: (i) a reference product $p+$ that maximally satisfies the user's preferences and (ii) a reference product $p-$ that maximally opposes them. For example, a reference product can be a theoretical or existing product that contain all ontology product tags related to a preference tag. The maximum possible positive aggregated association score product is defined as the aggregated association score of a product that contains all product tags positively associated with a preference tag:

$$\eta^{+}(\omega) = \sum_{z} r(z,\omega), \quad \{z \mid z \in \mathbb{z}_{p+} \wedge r(z,\omega) > 0\} \tag{B.17}$$

Such product is referred to as *positive reference product*, and its association score is the *positive reference association*.

Following the same principle, a product with the minimum possible negative aggregated association score is also introduced. Such product is referred to as *negative reference product*, and its association score is the *negative reference association*. The calculation for such score is:

$$\eta^{-}(\omega) = \sum_{z} r(z,\omega), \quad \{z \mid z \in \mathbb{z}_{p-} \wedge r(z,\omega) < 0\} \tag{B.18}$$

Once the aggregated association scores are calculated, a normalization is applied by dividing with the preference scores related to a preference tag. Therefore, comparison between different products is possible:

$$\eta^{*}(p,\omega) = \begin{cases} \dfrac{\eta(p,\omega)}{|\eta^{-}(\omega)|}\,, & \eta(p,\omega) < 0 \\ 0\,, & \eta(p,\omega) = 0 \\ \dfrac{\eta(p,\omega)}{\eta^{+}(\omega)}\,, & \eta(p,\omega) > 0 \end{cases} \tag{B.19}$$

Existence of overlaps between product tags when calculating the aggregated association of theoretical reference scores based causes the normalized aggregated association to approach 0, as the denominator in Relation (B.19) increases. Still, it is guaranteed that support or opposition towards a preference tag are not switched due to the overlaps in normalization, since the denominator is positive in both *Relations* B.17 and B.18. The normalized association score is bound in the range $[-1, 1]$, since an actual product may have at most the maximum positive or minimum negative aggregated association score towards a preference tag. Therefore, the term $\eta^{*}(p,\omega)$ is also bound in range $[-1, 1]$. The choice of the reference products is left to the ontology designer. Possible choices for a reference product are:

1. an existing product that achieves the highest/negative aggregate association score.
2. a theoretical or existing product that shares all positive/negative associations with a preference tag, while respecting the reduction design principle.
3. a theoretical or existing product that shares all positive/negative associations with a preference tag, without respecting the reduction design principle. This option calculates the highest possible denominator value in Relation (B.19). This clipping mechanism is illustrated in Figure B.1.

The above choices were evaluated with the app testers, and the normalized aggregated scores produced with the third choice were used during the ASSET field test.

Sustainability Index: Non-personalized Product Representations

The next step to calculate the product ratings is to establish a relationship between preferences and products. Aggregation over the normalized association score of a product and preference tags of a preference state estimates this relationship. This aggregation is performed by using a measure of central tendency as the estimator. More specifically, here the expected value is chosen. The expected value of the normalized aggregated association scores of all the preference tags that are related to the preference is an estimate that quantifies the support or opposition of a product by the preference:

$$v(p, c) = \frac{\sum_{\omega \in \Omega_c} \eta^*(p, \omega)}{|\Omega_c|} \tag{B.20}$$

Each preference tag represent a sustainability goal considered in a preference statement. When a product has positive normalized aggregated association score with a preference tag, then it supports in achieving the sustainability goal represented by that tag. In the opposite case, the product may cause the failure of achieving a sustainability goal. Therefore, the above calculation is referred to as the sustainability index of a product for a preference, as it indicates whether purchasing a product supports in achieving or failing a sustainability goal. Calculating the expected preference-product associations for all preferences and a product yields a numerical vector representation for that product in the preference semantic space. Such representations are used to compare products and determine whether a product is expected to be supported or opposed by a set of preferences. Measures of central tendency over a product representation, such as the mean $\overline{v(p, c)}$ can be used to calculate the non-personalized ontology estimate of the product sustainability.

Self-Determined User Personalization

High preference scores indicate that a preference is important for a user when calculating the product rating. Therefore, a user's preference scores are used as weights of importance for each preference. A user can express both opposition and support towards any preference. Thus, any inaccuracy or bias introduced by the ontology design may be mitigated by the user by adjusting preference scores. This

is also considered as an implicit user-determined extra correction on overlapping *primitive concepts*. Opposition and support of a user towards preferences are modeled via the offset of a preference score from the preference score median $\overline{s}$ as shown in Relation (B.21):

$$o(s_{u,c}) = s_{u,c} - \overline{s} \tag{B.21}$$

The higher the support or opposition of a user towards a preference, the higher the absolute value of the offset.

A weighted average between product-preference association scores and the user preference offsets are used to calculate a personalized association between a user and a product, given the user's self-determined preference scores. The sum of absolute offsets is used as denominator to preserve the sign of the rating while normalizing. Preserving the signs allows to extend the association logic to user-product level. Positive product ratings indicate that the product mostly supports the user supported preferences and opposes user opposed preferences. For negative product ratings, the product opposes user supported preferences and supports user opposed preferences. Although self-determined personalization allows for a user's *subjectivity* to influence the ratings, the user is made aware about which preferences produce such ratings. This results in a learning effect, that increases user awareness towards their sustainability preferences. For example when a person tunes the preferences to allow their favorite products achieve high scores, they are aware that to do so, they have to go against the preference they originally support. Thus the unscaled product rating is calculated as follows:

$$\varrho^*(p,u) = \frac{\sum_c v(p,c) \cdot o(s_{u,c})}{\sum_c |o(s_{u,c})|} \tag{B.22}$$

The scale of $[-1,1]$ can be transformed to any range of real numbers by applying a linear scaling with parameters α and β. Transforming the rating scale has several applicability scenarios. The rating can be scaled to different ranges to match the most preferred grading system of the country, where the algorithm is deployed [358]. Another possible usage of the scaling coefficients is to attribute rating for asymmetric perception of negative and positive rating values [359, 360]. For example, based on the work of Parguel et al. [360], negative values have a higher impact on user perception. In such case, a different lower scaling coefficients α, β can be used to reduce the impact.

$$\varrho(p,u) = \alpha \cdot \rho^*(p,u) + \beta \tag{B.23}$$

The product rating scale is designed to utilize user preference scores and create an association between a user and a product based on the product-preference associations. The rating value is expected to be bounded in a range $[\beta - \alpha, \beta + \alpha]$. Products that neither support nor oppose the user's preferences are assigned the mean value of the range β. If the product supports a user supported preference or opposes a user opposed preference, then the product rating increases. On alternative scenarios the product rating decreases.

The rating value compresses *overwhelming information* and shows to the user an estimate of the personalized product sustainability. Depending on the UI design, it is

possible to allow the user to further explore the ontology dynamics that result in this rating value. In the Figure B.2 all related ontology entities and rating calculations are presented.

Algorithm Complexity

All calculations to compute the product-preference representations rely on information that is not related to the user. Therefore, *Equations B.15-B.20* can be computed without privacy intrusion risks. The computational cost is significantly reduced for the user's device, since it is possible to calculate, store and distribute the product representations $v(p, c)$ by using a central database system. The calculation and storage complexity for the worst case scenario, where all products are connected to every product tag, all product tags are associated to all preference tags and all preference tags are connected to all preferences is:

$$O(n) = P \cdot T \cdot \Omega \cdot C \tag{B.24}$$

where:

P: The number of all products. T: The number of all product tags. Ω : The number of all preference tags. C : The number of all preferences.

Assuming that all the above sizes are equal, the worst case computational and storage complexity for the algorithm is polynomial to the power of 4:

$$(\text{Relation } \ B.24) \wedge (P = T = \Omega = C) = O(n^4) \tag{B.25}$$

Since the total products and product tags are often significantly more than the preference statements and preference tags, the expected time and space complexity is reduced to quadratic polynomial time $O(n^2)$. Therefore, modern CPUs on mobile phones and database servers can handle up to hundreds of thousands of product rating calculations per minute and store several thousands of products and product tags [361].

Overlaps and possible errors

As discussed in Section B.3, the application of the reduction design principle introduces several trade-offs between the efficiency of the rating calculations and maintenance of the ontology. Time constraints and limited resources regarding the construction and testing of the ontology may allow overlaps in the ontology, which introduce errors and biases that may break the bounding properties of *Equation* B.5. Thus, in such cases a clipping normalization is introduced to avoid numerical instabilities due to overflows outside the theoretical aggregated association bounds:

$$g(a) = \begin{cases} -\tau \ , & \text{if} \quad a \leq -\tau \\ a \quad , & \text{if} \quad -\tau < a < \tau \\ \tau \quad , & \text{if} \quad a \geq \tau \end{cases} \tag{B.26}$$

The introduction of the clipping changes *Relations* B.15, B.17 and B.18 to the following:

$$\eta(p,\omega) = g(\sum_{z\in\mathbb{z}_p} r(z,\omega)) \tag{B.27}$$

$$\eta^{+}(\omega) = g(\sum_{z} r(z,\omega)), \quad \{z \mid z \in \mathbb{z} \wedge r(z,\omega) > 0\} \tag{B.28}$$

$$\eta^{-}(\omega) = g(\sum_{z} r(z,\omega)), \quad \{z \mid z \in \mathbb{z} \wedge r(z,\omega) < 0\} \tag{B.29}$$

Even if overlaps affect the rating process, the users are able to mitigate the error by readjusting their preference scores, introducing an extra correction mechanism from their side. An illustration of the clipping mechanism on the normalized aggregate score is found in Figure B.1 in step 1.

Tractable and Explainable Ratings

The proposed ontology design and rating calculations rely on a fully tractable analytical framework. It is possible to calculate the exact amount that a preference, preference or product tag contributed to the rating. More specifically following Relations (B.22) and (B.23), one could solve to calculate the exact contribution of a specific preference c^* to the product rating as follows:

$$\varrho(p,u) = \alpha\left(\frac{\sum_{c\in\mathfrak{c}-\{c^*\}} v(p,c)\cdot o(s_{u,c})}{\sum_c |o(s_{u,c})|} + \frac{v(p,c^*)\cdot o(s^*_{u,c})}{\sum_c |o(s_{u,c})|}\right) + \beta \tag{B.30}$$

Therefore the contribution of preference c^* to the final rating is calculated as:

$$\alpha\frac{v(p,c^*)\cdot o(s^*_{u,c})}{\sum_c |o(s_{u,c})|} \tag{B.31}$$

The contribution of a specific preference tag ω^* is calculated following the same decomposition logic on Relations (B.20), (B.22) and (B.23), the contribution of a specific preference tag to the rating can be calculated as:

$$\alpha\frac{\sum_{c\in\mathfrak{c}} \frac{\eta^*(p,\omega^*)}{|\Omega_c|}\cdot o(s_{u,c})}{\sum_c |o(s_{u,c})|} \tag{B.32}$$

And the contribution of a specific product tag z^* to the product rating can be calculated by decomposing Relations (B.19), (B.20), (B.22) and (B.23):

$$\alpha\frac{\sum_{c\in\mathfrak{c}} \frac{\sum_{\omega\in\Omega_c} \frac{r(z^*,\omega)}{\eta_{norm}}}{|\Omega_c|}\cdot o(s_{u,c})}{\sum_c |o(s_{u,c})|} \tag{B.33}$$

where η_{norm} has the corresponding denominator value of either $\{\eta^+(\omega), \eta^-(\omega)\}$ that normalizes the aggregated association in Relation (B.19). In case that clipping is used, as shown in Relation (B.27), the contribution is calculated proportionally:

$$\alpha \frac{\sum_{c \in \mathfrak{c}} \frac{\frac{r(z^*,\omega)}{\sum_z r(z,\omega)}}{\eta_{norm}}}{\sum_c |o(s_{u,c})|} \cdot o(s_{u,c}) \tag{B.34}$$

Product tag contributions can also be calculated per preference or preference tag. Such calculations are possible by ommitting the calculation related to the product rating and calculate the amount a product tag contributes to other scores, such as the sustainability index discussed in Relation (B.20). Finally, contributions can either be stored when calculating the product rating, increasing memory complexity or recalculated after the rating calculation, thus increasing computation complexity.

Crowd Sourced Product Data

To facilitate wisdom of the crowds, a datathon took place at the University of Tartu, Estonia on November 2017 [362]. In total three teams of 3-4 people participated in the datathon, and produced data files and code within 72 hours. The datathon teams processed text data from Wikipedia pages related to product characteristics and sustainability goals. Each team used statistical methods and machine learning techniques based on NLP and sentiment analysis to extract an association score between product tags and sustainability topics. Each score should cite the wikipedia page id that was used to extract such data. This resulted in several thousand association scores between product and preference tags. The ASSET consortium used the majority of these results to evaluate existing association score and discover new ones.

Several new product tags and associations were introduced in the ontology, mainly in the environmental and social categories. Further collaboration with members of the winning datathon team took place in order to enrich and evaluate the created ontology.

As new product tags and association scores were discovered, the datathon provides a real-world example on how machine learning can be used to populate a sustainability ontology. Furthermore, the datathon showcases a working case, where crowd-sourced data can be used for populating a sustainability ontology.

Software Libraries

Data were collected, stored and processed for categorization and classification by AINIA according to GDPR. The smart phone app integrates a library version [363] of the Nervousnet system [364]. While no Nervousnet functionality has been used during the lifetime of this project and work, its integration has been part of a deliverable for

the ASSET EU-project. Its modular and flexible sensor data management system has potential future applicability in shopping scenarios from multiple retailer shops.

For the development and analysis of the presented results several libraries were used. Some notable references are: (i) pycausalimpact[2], (ii) scikit-learn [365], (iii) scipy [366], (iv) the discrete information theory package [367], (v)pyclustering [368], (vi) plotly [369]. Several other libraries that were used, are also mentioned in the respective code repository.

B.4 Supplementary Notes

Entry and Exit Survey

Entry and exit surveys are presented to the user during the field test. The aim of the surveys is to collect data regarding user opinions on sustainable consumption, sustainability preferences and the usage of the application. The number of participants for each survey is summarized in Table B.18 Answers on the exit survey provide useful insights regarding the behavioral change and sustainability awareness of users during the field test. Users confirmed that the rating methodology offers product ratings that justify their preference settings. Access to product information and the product tags is visible in the application for users that click on a product. Such information assists the decision or the product purchase. User declared that it is relatively easy to access the product information and that the provided information justifies the rating (*Supplementary Figures* B.5a and B.5b). The highly rated products tend to match the preferences of the majority of the users as shown in *Supplementary Figures* B.5c, B.5e and B.5f. Finally, an increase in discovery of novel products is stated by the users in Figure B.5d.

Shopping behavior is affected for some users by using the app. Mainly in retailer B users support that they would not buy products that achieve low ratings and buy high rated products before using the app, which emphasizes their sustainability focus (*Supplementary Figures* B.6a and B.6a). Regarding future purchases, users support that they will buy products that achieve high ratings and also avoid buying products that achieve low rating, as shown in *Supplementary Figures* B.6c and B.6d. Furthermore, users in retailer A estimated that they buy over 5% of highly rated products in retailer A, whereas most retailer B users estimate a value over 10%, as illustrated in Figure B.8. In both retailers, the majority of users agree to an extent that the usage of the application raised their awareness towards sustainability Figure B.6e

Product information and price seem to be the main decision factors when users decide regarding a product purchase, as illustrated in Figure B.7.

Concluding, weakly qualitative evaluation indicates that the application provides enough information to the users to purchase highly rated products, thus increasing the sustainability of their consumption. Screenshots from the app UI of the survey and preferences is illustrated in Figure B.11. Users perceive that the product rating

[2] `https://github.com/dafiti/causalimpact`

takes into account their preferences and is well justified by the available data. Other factors may affect the participation of users such as technical problems with the app, low usability of UI, seasonality (traveling), limited access to retailers shops for some consumers, etc.

Evaluation of Product Prices

Sustainability Index vs Product Price

This evaluation studies whether products that highly support a specific preference have a higher price compared to products from the same category. Evaluating sustainability index and price over all products provides insights about the general expenditure. Consumers might be interested in replacing products they buy with more sustainable products from the same category. To evaluate whether more expensive products achieve higher sustainability scores compared to competitive products from the same category, both price per unit and sustainability index are rescaled using the min-max normalization for each product per category. I.e. Each value is normalized by substracting the minimum value and then dividing by the difference between maximum and minimum value [370]. For the normalization to work, each category needs to contain at least two products with different sustainability index and price values. Therefore, only such categories are used for the analysis. As illustrated in Figure B.9a, purchasing more expensive products per category in retailer A seems to have almost no impact on the sustainability index for most categories. The vegan diet preference "H.12" is negative correlated with price, possibly indicating that the most expensive products per category often contain an animal product. For retailer B, the sustainability index for most preferences is either positively correlated or uncorrelated with prices. Negative correlated preferences with prices are "E.2", "E.3" and "S.1", indicating that one can purchase cheaper products per category, which are farmed in a sustainable manner, distributed in an environmentaly friendly way and passed sustainability auditing process. This result may indicate that in most categories the most expensive product originates from a place that the transportation has high environmental impact, sustainable farming techniques are not followed or sustainable auditing processes do not hold. Furthermore, since retailer B is a sustainability focused retailer, it is expected that the suppliers may mitigate sustainable production costs in their product prices.

Note that the lower expenditure level of Retailer A during the summer is a seasonality effect that is prominent due to the students' holidays and constructions in the neighborhood.

Open Data and Code

The ontology data is available here:
`https://doi.org/10.6084/m9.figshare.8313257`
The product rating algorithm is available here:
`https://github.com/asikist/value-sensitive-design-code`

Table B.28: (Max-Min) Normalized price per category and normalized sustainability index for retailer B. Relevant to Figure B.9b.

Preference	Correlation	p-value
E.1	-0.03	0.03
-E.2	-0.17	0.00
-E.3	-0.34	0.00
+H.1	0.23	0.00
+H.2	0.22	0.00
H.3	0.02	0.92
H.4	0.09	0.00
H.5	-0.09	0.00
+H.6	0.12	0.00
H.7	-0.01	0.59
H.8	-	-
H.9	-0.03	0.15
+H.10	0.40	0.02
H.11	0.01	0.78
+H.12	0.17	0.00
+H.13	0.20	0.00
Q.1	-	-
Q.2	-	-
Q.3	-	-
-S.1	-0.30	0.00
S.2	-	-
+S.3	0.29	0.00
+S.4	0.22	0.00
+S.5	0.35	0.00
S.6	-0.08	0.05

B.5 Nomenclature

Table B.30: Nomenclature Part I for Chapter 3.

u	A user, where $u \in \mathbb{u}$.
$\mathbb{u}$	A set of users.
T	A binary variable $\mathrm{T} \in \{0, 1\}$ indicating the participation of a user in a treatment.
R	A binary variable $R \in \{0, 1\}$ indicating the (possible) response of a user receiving a treatment
$\mathbb{u}_{\mathrm{T}=1, R=1}$	A set of users that received and responded to a treatment.
$\mathbb{u}_{\mathrm{T}=0, R=1}$	A set of users that did not receive but would probably respond to a treatment.
X	A vector/matrix/tensor of covariates that are used to estimate a treatment effect on a behavioral metric Y.
Y	A behavior metric that is used to measure the effect of a treatment.
p	A product, where $p \in \mathbb{p}$.
z	A product tag belonging to a product tag set $\mathbb{z}$.
$\mathbb{z}_p$	A product tag set for a specific product p.
c	A preference statement, where $c \in \mathbb{c}$.
ω	A preference tag belonging to a preference tag set $\mathbb{\Omega}$
$\mathbb{\Omega}_c$	A preference tag set related to a specific preference c.
$s_{u,c}$	A preference score given by a user u to a preference statement c.
$o(s_{u,c})$	The subtraction of the preference score median value $\widetilde{s}$ from a preference score $s_{u,c}$.
8705	The absolute value of a number $\cdot$ or the cardinality of a set.
8705	The absolute value of a number $\cdot$ or the cardinality of a set.
$\mathbb{Q}$	A set of concepts.
$\mathbb{Q}_\omega^+$	A set of concepts that support a preference tag ω.
$\mathbb{Q}_\omega^-$	A set of concepts that oppose a preference tag ω.
$\mathbb{Q}_\omega$	A set of all concepts that define a preference tag ω.
$\mathbb{Q}_z$	A set of concepts that define a product tag z.
$r^+(z, \omega)$	The positive association score between a product tag z and a preference tag ω, which is maximized when the concepts of a product tag z are enough to support and define the existence of a preference tag ω.
$r^-(z, \omega)$	The negative association score between a product tag z and a preference tag ω, which is maximized when the concepts of a product tag z are enough to support and define the absence of a preference tag ω.

Table B.32: Nomenclature Part II for for Chapter 3.

$r(z, \omega)$	The association score between a product tag z and a preference tag ω.
$r(z, \omega)$	The association score between a product tag z and a preference tag ω.
ϵ	The error introduced when aggregating tag associations $r(z, \omega)$ in the approximation of $\eta(p, \omega)$. The overlaps may be or on positive and negative concepts, causing the corresponding ϵ^+, ϵ^- errors.
$\eta(p, \omega)$	The aggregated association score between a product p and a preference tag ω.
$\eta^+(\omega)$	The positive reference association score that can be achieved for a preference tag c and all products $\forall\, p \in \mathbb{p}$. The negative reference correlation is referred as $\eta^-(\omega)$.
$\eta^*(p, \omega)$	The normalized aggregated association score for a preference ω and a product p.
$v(p, c)$	The average association value between a preference p and a product c.
$\varrho^*(p, u)$	The raw product rating assigned to a product p based on the preferences of a user u before scaling.
$\varrho(p, u)$	The product rating assigned to a product p based on the preferences of a user u.

Appendix

C

From Decentralized to Hybrid Basket Recommendations

C.1 Data Availability

Dunnhumby transactions and data are downloaded from Ref [91]. Food Agricultural Organization Food Balance sheets are taken from Ref. [92]. The environmental impact data are taken from Ref. [11]. Finally, all the data generated in this study are openly accessible in `www.github.com/asikist/g3/data`.

C.2 Code Availability

Code is available in Github `www.github.com/asikist/g3`

C.3 Fractional Decoupling

The personalized sustainable basket problem is presented as a mixed integer programming problem (MIP). Therefore, we need to consider discrete outputs for the proposed optimization schemes. NNs are known to operate in real value settings, as back-propagation requires the output of NNs to be continuous and differentiable in regards to objective, so that the chain rule can be efficiently applied. Unfortunately, this is not the case for MIP schemes.

In a typical NN feed-forward setting, we denote the output of the NN with weight vector $\boldsymbol{w}$ as $\boldsymbol{y}$ and its input $\boldsymbol{x}$. We are interested in gradient back-propagation when facing the integer output problem. More specifically when one could apply a floor rounding on output elements y_i by subtracting the fractional part $h_i = y_i - \lfloor y_i \rfloor$, i.e. the difference between y and the closest lower integer value $\lfloor y_i \rfloor$. When rounding to the lower closest integer, the NN output is transformed as follows:

$$f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{y} - \boldsymbol{h} \tag{C.1}$$

where, $\boldsymbol{h}$ is the vector of the fractional parts of $\boldsymbol{y}$. The main challenge for back-propagating error from the above operation is the calculation of the gradient of

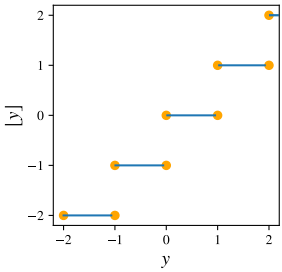


FIGURE C.1: A plot of the floor function for $x \in [-2, 2]$. Orange points indicate the regimes, where the floor function is not differentiable.

Relation (C.1) is that the floor function is not differentiable in its domain, in particular at the integer values (see Figure C.1).

A potential approach is to train the NNs in a real valued manner and then apply rounding when evaluating the solutions. In the specific case, this proves problematic, as many product quantities end up being rounded to 0, yielding empty baskets as solutions. Another approach proposed in literature is to use the Gumbel soft-max operator [371], which allows for gradient propagation via the so-called straight through estimators [372]. Since the decision problem in question requires no upper bounds on purchased product quantities, using the Gumbel soft-max operator may yield highly dimensional outputs that may take considerable more time to train for large scale problems.

An alternative approach, termed fractional decoupling, is proposed to efficiently calculate a gradient update and perform gradient descent. To perform fractional decoupling, we simply subtract the fractional part h_i of a real valued output y_i, while treating it as constant, i.e. we allow no gradient propagation through it in the computational graph, similar to bias terms or physics inspired NNs constants.

Proposition 2. *Decoupling the fractional part h of real value $y \in \mathbb{R}$ from the gradient calculations allows for feed-forward rounding and well-defined gradient propagation when the loss function is differentiable w.r.t. to y in* $[\min(y-h), \max y_i]$

Proof. The fractional part $\boldsymbol{h}$ is considered constant, that is no gradient propagation is allowed through it when calculating gradients. For a network parameter w_k, the gradient calculated after applying the above operation will have the following form

$$\frac{\partial J(y-h)}{\partial w_k} = \frac{\partial J(y-h)}{\partial (y-h)} \frac{\partial y - \partial h}{\partial w_k}. \tag{C.2}$$

Since h is treated as a constant $\partial h = 0$

$$\frac{\partial J(y-h)}{\partial w_k} = \frac{\partial J(y-h)}{\partial (y)} \frac{\partial y}{\partial w_k} = \frac{\partial J(y-h)}{\partial w_k}. \tag{C.3}$$

The resulting gradient corresponds to the original gradient of the loss function shifted to the nearest integer NN output $y - h$, which is well defined given the theorem assumption. □

The above proof can be respectively extended to any rounding operator, such us ceiling or half-up. The introduced fractional decoupling technique effectively allows gradient flows through discretization operators in order to facilitate learning.

Numerical Example of Discretized Gradient Flows

To illustrate the gradient discretization of fractional decoupling we provide an illustrative example, which compares gradient flows of a 2-parameter network both with and without fractional decoupling when solving the same problem. Given a predetermined coefficient vector $\hat{\boldsymbol{w}} = [\hat{w}_1 \quad \hat{w}_2] = [4 \quad 1]$ and normally distributed inputs $x_1, x_2 \sim \mathcal{N}(\mu = 0, \sigma = 1)$, apply the following transformation is applied:

$$\hat{\boldsymbol{y}} = \lfloor \hat{\boldsymbol{w}} \odot \boldsymbol{x} \rfloor \tag{C.4}$$

Given a batch matrix of $M = 50$ input row vectors $X \in \mathbb{R}^{M \times N}$ with $N = 2$ features columns and their corresponding label matrix $\hat{Y} \in \mathbb{R}^{M \times 1}$ we train two NNs. A NN with continuous outputs during training, where the floor function is applied only during inference:

$$\boldsymbol{f}_1(\boldsymbol{x}) = \boldsymbol{y} = \boldsymbol{w} \odot \boldsymbol{x} \tag{C.5}$$

and a separate NN with a fractional decoupling term:

$$\boldsymbol{f}_2(\boldsymbol{x}) = \boldsymbol{y} - \boldsymbol{h}. \tag{C.6}$$

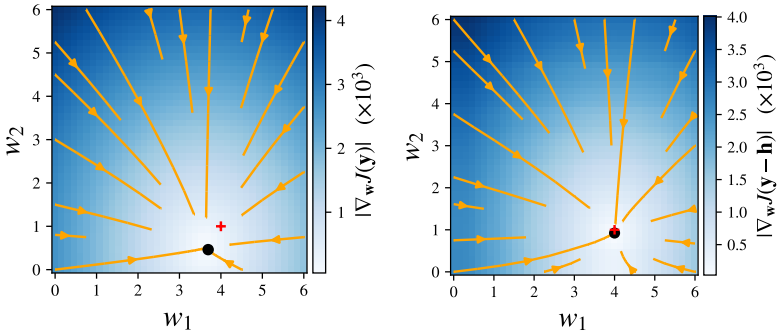
The mean squared loss is minimized during learning over a single batch of $M = 50$ samples:

$$J(\boldsymbol{f}_\cdot(\boldsymbol{x})) = \frac{1}{M} \sum_1^M \frac{1}{2} \left\| \boldsymbol{f}_i(\boldsymbol{x}) - \hat{\boldsymbol{y}} \right\|_2^2 . \tag{C.7}$$

For the experiments, 40 weight values evenly spaced in the interval $[0, 6]$ are considered for each NN parameter $w_1\, w_2$ and generate all possible pairs (cartesian product). For each parameter vector of $\boldsymbol{w} = (w_1\, w_2)$ the gradients $\nabla_{\boldsymbol{w}} J(f_1)$ and $\nabla_{\boldsymbol{w}} J(f_2)$ are calculated respectively. The calculated gradients are illustrated in Figure C.2, where it is observed that fractional decoupling approximates the underlying coefficient vector better. In general it is observed that using fractional decoupling has little effect on the gradient direction and magnitude. The minimum magnitude gradient points for each NN are expected to be a local minima for J(see black disks in Figure C.2). During inference, it is observed that fractional decoupling yields a lower loss value $J(\boldsymbol{y} - \boldsymbol{h})) \approx 0.02$ compared to the continuous NN with floored output $J(\boldsymbol{y})) \approx 0.28$.

C.4 Gradient Guided Genetics

Technical details for gradient guided genetics are found in this section. Each element of the gradient guided genetics depicted as a module Figure 4.1 is explained in more detail in the following sections.



(a) Continuous gradient.

(b) Fractional decoupling gradient.

FIGURE C.2: Gradient direction (orange lines), optimal solution (red cross) and lowest gradient norm point (black disk).

Neural Crossover

For neural crossover we consider the use of transformers that use the attention mechanism [272]. A typical transformer architecture receives two sequences $x_1 \in \mathbb{R}^N, x_2 \in \mathbb{R}^M$ as an input and outputs a real valued (logit) matrix $A = \mathbb{R}^{N \times M}$ that can be transformed to a probability matrix by applying a soft-max operator over its rows. The probability vector denotes the "attention" or assigned probability that an element from the second input sequence x_2 is important to match against the corresponding element i from sequence x_i. Transformers are the state-of-the-art models used in several architectures that perform sequence matching related learning tasks, such as machine translation and text generation [373]. Often, multi-head attention mechanisms are used, where each "head" represents different NN layers that focus "attention" on different parts of each sequence. In the current application, transformers are assumed to be able to naturally learn how to match parent chromosomes to optimize the goals. The output probability matrix is used to make a weighted average calculation between matched chromosomes as shown in Relation (4.10).

The multi-head attention transformer networks used in this work contain 1 encoder and 1 decoder layer with GeLU activation functions and 11 heads. Replacing the crossover network with probabilistic operators or simpler NN architectures has not yielded better results so far, but is still a subject of study and future work. Both of the transformer encoder and decoder layers contain a hidden layer with 2048 hidden neurons and, layer norm layers in output and input and also dropout operations on neuron outputs, according to the default implementation found in `https://pytorch.org/docs/stable/generated/torch.nn.Transformer.html` (accessed October 2021). A sigmoid activation is used on the attention output to convert it in the range between $[0, 1]$.

Neural Mutation

The neural mutation operator follows the method proposed in Relation (4.11). The evolution period is $[0, 1]$ and a single hidden layer with 256 neurons for both decentralized and hybrid designs. In total 9 mutated solutions are selected per trajectory. In the current setting, this parameterization yields high performing results, but in problem different settings, higher mutation period value and other activation functions might work better. Replacing the mutation network with probabilistic operators or simpler NN architectures has not yielded better results so far, but is still a subject of study and future work.

Population Discretization

Discretization of quantities in chromosomes is performed via fractional decoupling described in Section C.3. A ReLU function is applied to the real-valued chromosomes to preserve non-negative quantities. Then a fractional decoupling rounding is performed on the outcome. The back-propagation of fractional decoupling is confirmed by observing the parameter changes and loss convergence over time in Figure C.3.

Back-Propagation Through Evolution

The back-propagation through evolution starts by calculating the individual objective function values for each solution selected by the selection operator. For each objective, the mean value over all selected individuals is calculated. Experimental results indicate that using a different optimizer for each Neural Operator yields higher performance. The RMSProp optimizer is used with learning rate $\eta = 0.0001$ for the neural mutation operator and an RMSProp optimizer with learning rate $\eta = 0.0001$ for the Neural Crossover Operator. The gradient is calculated per objective, and for health objectives the loss is scaled 7 times. Not scaling the loss yielded recommendations that did not optimize health objectives well.

A "stopping criterion" is calculated in the following manner in each generation. as the mean over all non-dominated solution objectives. The "stopping criterion" value is preserved for last three generations, and the mean value of the current generation is compared against the mean value of the last three generations "stopping criteria". If the current "stopping criterion" differs less than 0.001 from the mean of the previous "stopping criteria", the optimization is terminated. Learning convergence of the total loss and individual losses are found in Figure C.3.

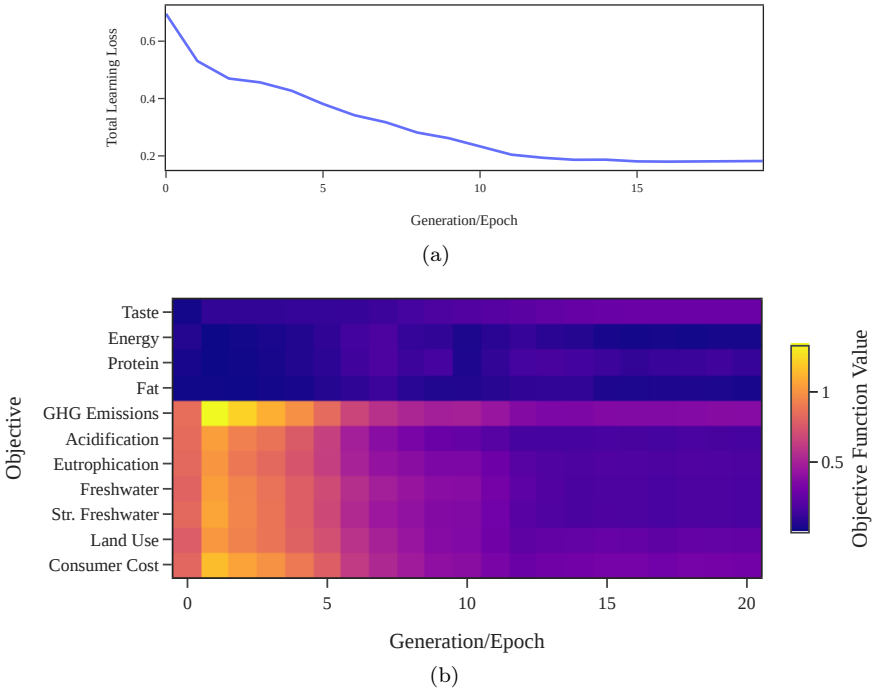


FIGURE C.3: Learning loss (line) and normalized median values of the separate objective functions (heatmap) over the non-dominated chromosomes of each generation, for a single basket recommendation. As generations evolve, G3A converges all losses and the learning objective to lower (darker blue) values. If no convergence criterion is used, G3A may over-fit and produce solutions that optimize all objectives except Taste. This behavior was also observed in other baselines, and thus the cosine similarity filter was used.

Nomenclature

Table C.1: Nomenclature part I for Chapter 4

N	Number of distinct products and chromosome/solution length for the genetic.
$\boldsymbol{x}$	A non-negative integer vector $\boldsymbol{x} \in \mathbb{Z}_0^N$ representing a basket of product quantities.
$\boldsymbol{x}^*$	A purchased/intended basket used as the target basket for optimizing consumer taste.
j	A decision criterion/feature quantity.
$c_{i,j}$	A product feature quantity (e.g. kg of emissions) per unit of product i and feature j.
$v_j(\boldsymbol{x})$	The total quantity for a feature j quantity per basket.
$\boldsymbol{x}'$	A non-negative integer vector $\boldsymbol{x} \in \mathbb{Z}_0^N$ representing another basket of product quantities.
$\rho_j(\boldsymbol{x}, \boldsymbol{x}')$	The ratio of total feature quantities $v_j(\boldsymbol{x}), v_j(\boldsymbol{x}')$ between two baskets $\boldsymbol{x}, \boldsymbol{x}'$.
$\|\| \cdot \|\|$	The L2 norm of a vector.
x_i	Quantity of product i in basket j.
$J_j(\boldsymbol{x})$	The objective loss calculated for a basket $\boldsymbol{x}$ and objective j. More than one inputs may be provided.
C	The set of criteria j with cardinality $\|\mathrm{C}\| = M$.
X	The set of solutions/baskets $\boldsymbol{x}$. Also termed as population.
$\hat{\mathrm{X}}$	The feasible set of solutions/baskets $\boldsymbol{x}$, i.e. solutions that exist.
$\hat{\mathrm{X}}_0$	The initial set of solutions/baskets $\boldsymbol{x}$, i.e. solutions that are further optimized to generate recommendations. The generation can also be used as index to the population.
$\boldsymbol{x} \prec \boldsymbol{x}'$	An operator denoting that solution/basket $\boldsymbol{x}$ dominates $\boldsymbol{x}'$. The dominance operator can also be used to describe dominance for points of the objective space $\boldsymbol{\zeta}(\boldsymbol{x}) \prec \boldsymbol{\zeta}(\boldsymbol{x}')$, which typically represent solutions.
$\boldsymbol{\zeta}(\boldsymbol{x})$	A vector containing all objective values for solution $\boldsymbol{x}$ and indexed by j.
B	The population or solution set size.

Table C.3: Nomenclature part II for Chapter 4

$\boldsymbol{w}$	A parameter vector that is optimized to generate solutions.
f	An optimization algorithm that generates a set of solutions X given an input set of solutions tsx_0 and parameters $\boldsymbol{w}$.
δ	A probability sample to decide whether random crossover should happen.
p	A probability threshold to decide whether random crossover should happen.
f	Probability density function for sampling continuous values in $[0, 1]$.
κ	Value of discrete random mutation.
f_{discrete}	probability distribution function for sampling discrete values in κ.
F_α	The non-dominated frontier of rank $alpha$. F_1 contains all the non-dominated solutions of a population.
$\boldsymbol{X}$	A population represented as a matrix.
$\boldsymbol{g}$	An attention vector for an element x_i, representing the attention values assigned to all elements x_i' in the population.
b	The index of the second parent that received the highest attention.
$\boldsymbol{u}(\boldsymbol{x}(t))$	The NODEC output that controls neural mutation.
T	The total time that continuous mutation is applied on a chromosome.
τ	The generation index, that corresponds a specific generation in genetic optimization process. Here we represent the index at the total.
h_i	The fractional part of a real value $y_i \in \mathbb{R}$
$\lfloor y_i \rfloor$	The floor operation on a real value $y_i \in \mathbb{R}$.
J	The learning loss of a NN.
w_k	A NN parameter.

Table C.5: Nomenclature part III for Chapter 4

q	The week index.
k	The household index.
ν_k	Number of persons in household k.
ϵ	Noise for privacy-preserving masking, sampled from a Laplace distribution.
$\mathcal{L}(\mu, \beta)$	A Laplace distribution with location parameter μ and scale parameter β.
$\overline{v}_{j,k,q}(\boldsymbol{x}^*_{k,q})$	The mean feature quantity per person for a basket.
$m_{j,k,q}$	The mean masked feature quantity per person for a basket.
$\overline{m}_{j,q}$	The mean masked feature quantity per person for a basket.
K_q	Number of available households at week q.
σ	denotes the variance of a normal distribution, and in the MO-NES case the corresponding parameter.
$\boldsymbol{A}$	The matrix that represents a "covariance" related parameter in MO-NES.
$\boldsymbol{Z}$	A sample vector, where each element is sampled from a zero-mean unit variance normal distribution. This vector is used in MO-NES.
η	is used for learning rate parameter. In MO-NES, three learning rate parameters are used for each parameter, namely: $\eta_\sigma^+, \eta_\sigma^-, \eta_A$.

Other Baseline Algorithms

RNSGA-II

A non-dominated sorting algorithm may produce a large number of non-dominated solutions that are not preferable, e.g. solutions that optimize a single objective very well and not the others. To keep the population size B per generation constant, a secondary selection operation needs to be performed. Random selection is often undesired in problems that have multiple objectives [261], and thus a more sophisticated technique is preferred. Some probabilistic evolutionary algorithms use a sorting operation to perform a secondary selection operation that guide the evolutionary processes towards preferred non-dominated solutions, e.g. non-dominated that optimize specific combinations of the objectives very well. A typical example that will be used as a baseline in the current study is reference point NSGA-II, abbreviated as RNSGA-II [261], which uses reference directions to guide evolution towards preferred solutions. In brief, one or more reference points are selected to guide the evolution. A reference point $\hat{\boldsymbol{\zeta}}$ is generated by providing a vector of preferred objective values to the system. Each candidate solution receives two ranks determined by the non-dominated sorting and a distance metric from each reference point, i.e. lower distance values receive lower. Lower ranks are used to select the candidates for the next generation. This algorithm shows higher performance gains compared to NSGA-II to perform better on multi-objective problems with more than 2 objectives [261]. RSNGA-II is used as a baseline in the current study following the default implementation of [374].

A logistic map [266] is applied on the initial basket to generate the initial solution for RNSGA-II, improving performance considerably compared to other random initialization. Several reference points settings are tested for RNSGA-II. The current reference points provided to RNSGA-II are three, one that is calculated by using the infeasible optimum, where every loss is 0, one that minimizes all individual losses (e.g. all values for $j \leq 5$ are 0 and the rest are 1), and one that minimizes all environmental losses (e.g. all values for $j > 5$ are set to 0). Using less than 2 reference points resulted often in bad performance. Other reference point settings were tested on 100 intended baskets, such as using the one the intended basket or minimization of specific losses on smaller samples, but it was unclear whether better performance could be achieved by using them. The current reference point setup was chosen, as it provided the best performing dominance ratio when comparing to other baselines. Integer exponential crossover and polynomial mutation are used for the genetic operators. Finally, other settings were tested with $B = 100$, but were omitted the due to lower dominance ratio, slower convergence times, large number of solutions, and difficulty to determine subsets of good solutions.

MO-NES

Another way to handle multi-objective optimization problems is the use of Multi-Objective Natural Evolution Strategies (MO-NES) [260], which use a gradient guided search algorithm to find non-dominated solutions by parametrizing a probabilistic

model (relies on sampling). The algorithm optimizes the parameters of a model that samples solutions from underlying distributions. For each solution, a sample vector $\boldsymbol{z} \in \mathbb{R}^N$ is generated, where each element is sampled from a normal distribution $z_i \sim \mathcal{N}(0,1)$. A new solution $\boldsymbol{x}'$ is calculated based on a parent solution $\boldsymbol{x}' = \boldsymbol{x} + \sigma \boldsymbol{A}\boldsymbol{z}$, where $\sigma \in \mathbb{R}, \boldsymbol{A} \in \mathbb{R}^{N \times N}$ are the co-variance related terms. Samples from the previous population X_τ and the new candidates $\boldsymbol{x}'$ are combined into an intermediate population X$'$. Each solution $\boldsymbol{x} \in$ X$'$ is assigned a rank α based on the non-dominated sorting. A secondary rank β is assigned to each solution based on the value of a hyper-volume metric [260, 375] in a descending order. To calculate the hyper-volume metric, a dominated reference point $\boldsymbol{\zeta}^{(0)} \in \mathbb{R}^M$ is selected in the objective space, such that all considered solutions $\boldsymbol{x} \in$ X$'$ dominate this point $\boldsymbol{\zeta}(\boldsymbol{x}) \prec \boldsymbol{\zeta}^{(0)}$. The hyper-volume metric [375] is used to calculate the hyper-volume between each solution and the dominated reference point, e.g. by using the proposed implementation of Ref. [376]. The hyper-volume metric is calculated on normalized loss values, which are calculated by subtracting the mean and then dividing with the standard deviation over all solutions. The covariance related parameters $\boldsymbol{A}, \sigma$ are updated with a gradient update. A modified version of MO-NES, where solutions are rounded and negative values are clipped to 0 prior to evaluation is used as a baseline in the current chapter. The initial value of each solution is sampled as $x_i = \mathrm{ReLU}(x), x \sim \mathcal{N}(0, 0.2)$. Parameter $\sigma = 1/3$ and elements of A were initialized uniformly in $[0, 0.001]$. Following notation from Ref. [260], the learning rates for each parameter are $\eta_\sigma^+ = 0.01$, $\eta_\sigma^- = 0.01/5$ and $\eta_A = 0.01/4$. MO-NES trains up to 40 generations.

C.5 Dataset

Transaction data, product prices, and purchased quanties were retrieved by "The Complete Journey Dataset" by the Dunnhumby grocery store [91]. The quantities are included in US imperial units and a conversion to metric system was done in the following manner: (i) Unit labels are identified and grouped together with regular expressions, e.g. "LB,lb, LBs" all represent the same label which denotes pounds. (ii) Weight and volumetric labels are separated and proper conversion coefficients are used to convert each unit to the corresponding metric unit used in the other datasets. (iii) Prices may change through time, so the median price per unit is calculated through time and over all stores to generate the price features used by all algorithms.

Environmental impact indicators for product types are taken from Ref. [11]. Nutrition information from Food Agricultural Organization Food Balance Sheets [262] are downloaded from Ref [92]. All three datasets contain different product type labels for each product. From "The Complete Journey Dataset" the "SUB_-COMMODITY_DESC" column is treated as the product identifier. Each value of the column "SUB_COMMODITY_DESC" is matched against the "product category" column from FAO FBS dataset and the dataset column "product category" from Ref [92]. The resulting dataset contains transaction prices, purchased quantities, environmental impact values, and nutritional info per transaction. Any other dataset

with similar structure and more diverse product characteristics is compatible with the proposed method for finding more sustainable personalized baskets.

Appendix

Enabling Hybrid Design Approaches: Self-Determined privacy-preserving Collective Aggregations of Individual Data

D.1 Nomenclature

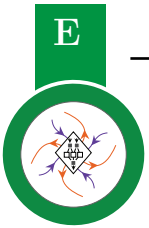
Table D.1: Nomenclature Part I for Chapter 5.

A	A multi-set of real values. Any capital letter is treated as a multi-set of real values, unless stated otherwise.
$g(A)$	a function that aggregates all elements of a set A into a real value. e.g. for sum $g_{sum}(A) = \sum_{i=0}^{\lvert A \rvert} a_i$
$\overline{A}$	The mean value of all elements of a set, where $a_i \in A$.
$\widetilde{A}$	The median value of all elements of a set, where $a_i \in A$.
$max(A)$	The maximum value of all elements of a set, where $a_i \in A$.
$min(A)$	The minimum value of all elements of a set, where $a_i \in A$.
$H(A)$	The Shannon's entropy value for all elements of a set, where $a_i \in A$.
a^*	A sub-optimal value that approaches an optimal value, e.g. $a^* \rightarrow max(A)$ or $a^* \rightarrow min(A)$.
a^*	A new sub-optimal value that approaches an existing sub-optimal value a^*.
n	A user.
N	A set of users.
t	A time index.

Table D.3: Nomenclature Part II for Chapter 5.

$s_{n,t}$	A sensor value generated in time t by the user n.
η	a masking mechanism, which consists of a parametric algorithm that masks the sensor values of a multi-set S.
$\theta_{\eta,k}$	A parameterization k for a masking mechanism η.
υ	A uniformly distributed variable.
$f_\eta\left(S,\theta_{\eta,k}\right)$	a privacy setting consisting of a masking mechanism η parameterized with parameters $\theta_{\eta,k}$ and operating on a set of sensor values S. It produces a masked set of sensor values $f_\eta\left(S,\theta_{\eta,k}\right)=M$, such that $\lvert S\rvert=\lvert M\rvert$.
Q	A multi-set of privacy values.
α_i	A parameter that weights the importance of privacy factors for calculating the privacy values.
δ	A parameter that denotes the amount of privacy that the data producer sacrifices or gains over the existing privacy.
c	A parameter that denotes the amount of utility that a data consumer sacrifices or gains over the existing utility value.
U	A multi-set of utility values.
α_i	A parameter that weights the importance of privacy factors for calculating the utility values.
γ_i	A parameter that weights the importance of utility factors for calculating the utility values.

Appendix



How Artificial Neural Networks Learn to Control Dynamical Systems

E.1 Data Availability

Data supporting this study and all source codes and ANN architectures are publicly available at `https://github.com/asikist/nnc`.

E.2 Directed Networks

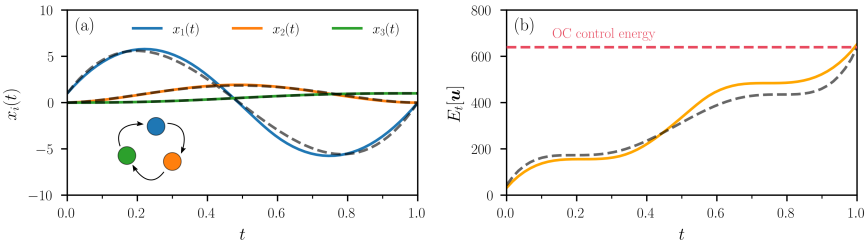


Figure E.1: **Controlling a connected 3-node cogwheel system. (a)** Evolution of system states $x_i(t)$ $(1 \leq i \leq 3)$ under OC (dashed lines) and neural-network-based controls (solid lines). **(b)** Evolution of the control energy $E_t[\boldsymbol{u}]$ for AI Pontryagin after 20,000 training epochs (solid orange line) and optimal control (dashed black line). The learning rate is $\eta = 6 \times 10^{-4}$.

This section discusses three different examples of directed networks which we control with AI Pontryagin.

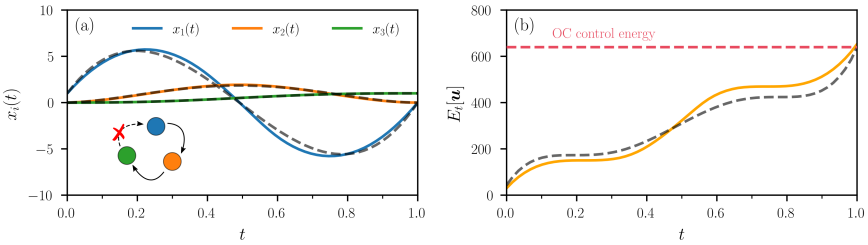


FIGURE E.2: **Controlling an unconnected 3-node cogwheel system. (a)** Evolution of system states $x_i(t)$ $(1 \leq i \leq 3)$ under OC (dashed lines) and neural-network-based controls (solid lines). **(b)** Evolution of the control energy $E_t[\boldsymbol{u}]$ for AI Pontryagin after 20,000 training epochs (solid orange line) and optimal control (dashed black line). The learning rate is $\eta = 6 \times 10^{-4}$.

As a first example, we study a 3-node cogwheel system with adjacency and driver matrices

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} . \tag{E.1}$$

Observe that the circulant matrix A satisfies $A^3 = \mathbb{1}$, so the spectrum of A is $\lambda(A) = \{1, e^{-2\pi \mathrm{i}/3}, e^{2\pi \mathrm{i}/3}\}$. The same holds for the transpose $A^\top$. Since A, $A^\top$ are diagonalizable, we can directly evaluate the corresponding matrix exponential in the controllability Gramian $W(t)$. Figure E.1 shows the evolution of $x_i(t)$ $(1 \leq i \leq 3)$ for the system that is subject to optimal control (OC) and AI Pontryagin-based control signals. The initial and target states are $\boldsymbol{x}(0) = (1, 0, 0)^\top$ and $\boldsymbol{x}^* = (0, 0, 1)$, respectively. We observe that both control approaches reach the desired target state for $T = 1$. The control energy associated with AI Pontryagin also resembles that of OC. The relative difference at time $T = 1$ is about 2%.

The ANN that we use to control the 3-node system consists of a single hidden layer with 20 neurons and an exponential linear unit (ELU) activation. We transform the hidden layer output to a control signal via a linear layer with 1 neuron that describes the single control input. We initialize the ANN weights $\boldsymbol{w}$ with the Kaiming uniform initialization algorithm [334]. For the gradient descent in $\boldsymbol{w}$ (Relation (6.4) in the main text), we use a learning rate $\eta = 6 \times 10^{-4}$. The number of time steps is 60.

As a second example, we consider the following directed system with an absorbing state and a non-invertible adjacency matrix

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} . \tag{E.2}$$

Observe that A and $A^\top$ are nilpotent matrices with degree 3. The matrix exponentials arising in the controllability Gramian can be directly calculated from their series expansions since the series have non-zero terms only for A^k with $k < 3$. For the same initial and target states as in the previous example, we show the evolution of $x_i(t)$ $(1 \leq i \leq 3)$ and $E_t[\boldsymbol{u}]$ in Figure E.2.

In the third example, we focus on a directed growing network [377] with 1,024 nodes. We select different proportions of controlled nodes uniformly at random. If 95% of all nodes are controlled, OC solutions cannot be calculated analytically since the controllability Gramian $W(T)$ (Relation (6.7) in the main text) is not invertible, implying that the linear dynamical system is not controllable. For larger proportions of controlled nodes, the OC approach gains in its ability to control the directed network. AI Pontryagin is able to minimize the loss function and steer the system very close to the desired target state for both examples shown in Figure E.3. The relative difference between the control energies of AI Pontryagin and OC for the solution shown in Figure E.3 is about 20%. In this example, node states are initialized with values uniformly sampled from the interval $[0, 1]$. The target state is generated by applying update rule Relation (E.4) to the initial state for 40 iterations. The ANN that we use to control the directed network consists of a single hidden layer with 15 ELUs. We used a learning rate $\eta = 1.2$.

E.3 Noise Robustness

To study the robustness of the backpropagation of loss function gradients during training, we carry out additional numerical experiments with additive noise that acts as uncertainty on the observed reached state at time T,

$$\hat{\boldsymbol{x}}(T) = \boldsymbol{x}(T) + \boldsymbol{\epsilon}\,, \tag{E.3}$$

where $\boldsymbol{x}(T)$ denotes the unperturbed reached state and $\boldsymbol{\epsilon}$ is the vector whose elements are distributed according to a Gaussian distribution $\mathcal{N}(0, \sigma)$ with zero mean and variance σ^2. The uncertainty associated with the observed reached state acts as a perturbation on the loss function and its gradients. If the signal to noise ratio is not large enough, gradients still carry enough information for efficient learning of control signals (see Figure E.4). If we introduce adaptive learning rates during the training process, it is possible to better control the level of target noise (see Figure E.5).

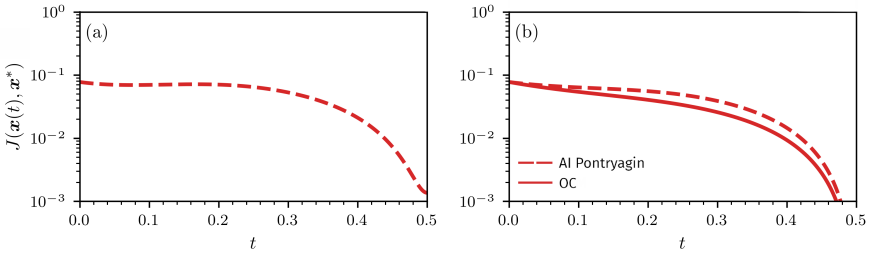


FIGURE E.3: **Controlling a larger-scale directed network.** We use AI Pontryagin to control a growing network [377] (directed graph) with 1,024 nodes for two different fractions of controlled nodes that are selected uniformly at random. **(a)** If 95% of all nodes are controlled, the controllability Gramian in Relation (6.7) (see main text) is non-invertible and OC solutions Relation (6.6) (see main text) cannot be calculated. AI Pontryagin still manages to steer the dynamics towards the target state and minimize the final loss $J(\boldsymbol{x}(T), \boldsymbol{x}^*)$. **(b)** For 99% controlled nodes, both methods minimize the loss function. Dashed and solid lines indicate AI Pontryagin and OC solutions, respectively.

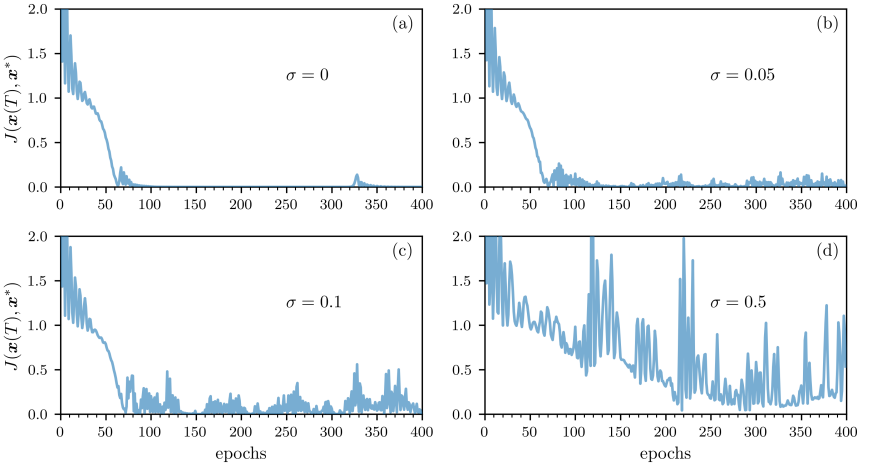


FIGURE E.4: **Effect of noise on learning performance of AI Pontryagin with fixed learning rate.** For the two-state system Relation (6.8) (see main text) and different noise levels ($\sigma = 0$ in **(a)**, $\sigma = 0.05$ in **(b)**, $\sigma = 0.1$ in **(c)** and $\sigma = 0.5$ in **(d)**), we show the loss Relation (6.5) (see main text) as a function of the number of training epochs. Gaussian noise with zero mean and variance σ^2 acts on the observed reached state $\boldsymbol{x}(T)$ according to Relation (E.3). We set the learning rate to a value of $\eta = 0.1$ and use the Adam optimizer.

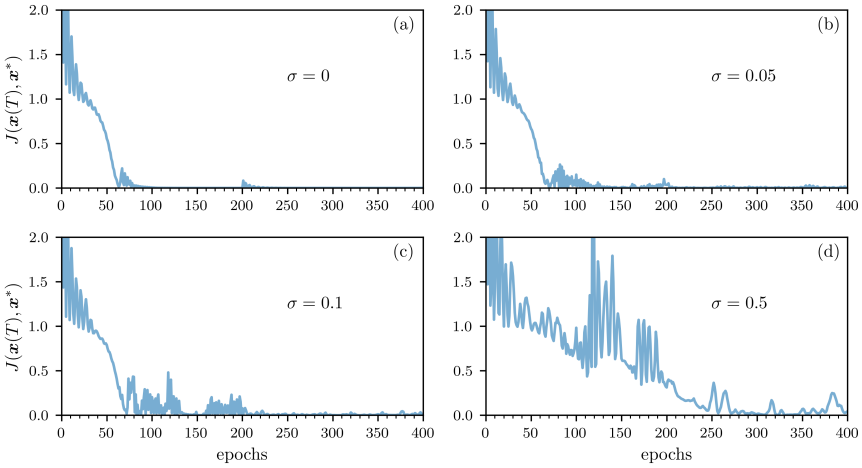


FIGURE E.5: **Effect of noise and adaptive learning rates on learning performance of AI Pontryagin.** For the two-state system Relation (6.8) (see main text) and different noise levels ($\sigma = 0$ in **(a)**, $\sigma = 0.05$ in **(b)**, $\sigma = 0.1$ in **(c)** and $\sigma = 0.5$ in **(d)**), we show the loss Relation (6.5) (see main text) as a function of the number of training epochs. Gaussian noise with zero mean and variance σ^2 acts on the observed reached state $\boldsymbol{x}(T)$ according to Relation (E.3). We initially set the learning rate to a value of $\eta = 0.1$. After 200 training epochs, we set $\eta = 0.01$ as a fine-tuning mechanism. Computations were performed with the Adam optimizer.

E.4 Driver Nodes

Kuramoto Dynamics

To study the influence of different fractions of controlled (*i.e.*, driver) nodes on the control performance of AI Pontryagin, we consider an Erdős–Rényi and a Watts–Strogatz network with $N = 225$ nodes (see Figure 6.4 in the main text) and different numbers of uniformly at random selected driver nodes. Neural-network architectures are as reported in the main text.

Figure E.6 shows the order parameter $r(T)$ (Relation (6.20) in the main text) for $T = 5$ and different driver node fractions. The coupling constant K is set to 10, 15, and 20% of the critical coupling constant K^*. For both networks, we observe that the order parameter is between 0.97 and 1 for driver-node fractions larger than 90%. For a fraction of controlled nodes of 95%, the order parameters are 0.987 ($K = 0.1K^*$), 0.990 ($K = 0.15K^*$), and 0.992 ($K = 0.2K^*$) in the Erdős–Rényi network, and 0.983 ($K = 0.1K^*$), 0.988 ($K = 0.15K^*$), and 0.992 ($K = 0.2K^*$) in the Watts–Strogatz network. For smaller fractions of controlled nodes, Figure E.6 shows that the order parameter in the Erdős–Rényi network is more sensitive to the fraction of controlled nodes than in the Watts–Strogatz network. These findings are

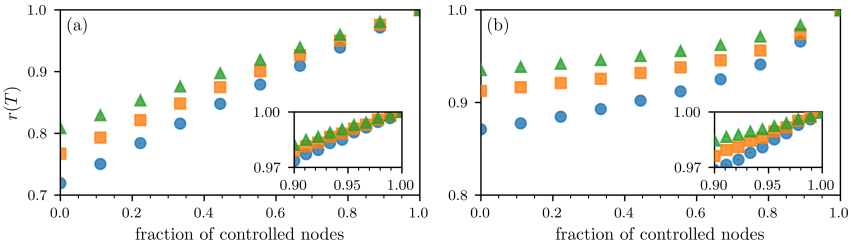


FIGURE E.6: **Dependence of the order parameter on the fraction of driver nodes.** The order parameter $r(T)$ (Relation (6.20) in the main text) of Kuramoto oscillator dynamics as a function of the fraction of controlled nodes for **(a)** an Erdős–Rényi network $G(N, p)$ with $p = 0.3$ and **(b)** a Watts–Strogatz network with degree $k = 5$ and rewiring probability $p = 0.3$. Both graphs have $N = 225$ nodes. Hence, the mean degree of the Erdős–Rényi network is 67.5. We set $T = 5$ and used learning rates η between 0.1 and 1.0. The shown results are based on averages over 1,000 i.i.d. natural frequency realizations. Different markers indicate different coupling constants: $K = 0.1K^*$ (blue disks), $K = 0.15K^*$ (orange squares), and $K = 0.2K^*$ (green triangles).

well-aligned with those reported in [77], indicating that in oscillator networks with a large mean degree and a small coupling constant, a large proportion of nodes need to be controlled to achieve a certain degree of synchronization. The order parameter in networks with smaller mean degrees and a subcritical coupling constant is less sensitive to the fraction of controlled nodes.

Linear Dynamics

Next, we discuss how linear systems can be controlled with a subset of all nodes (*i.e.*, driver nodes). We set A in $f(\boldsymbol{x}, \boldsymbol{u}) = A\boldsymbol{x} + B\boldsymbol{u}$ (see main text) equal to the adjacency matrix of a lattice graph with $N = 1,024$ nodes. To determine the driver matrix B and driver nodes, we use the maximum matching method [129]. The corresponding fraction of driver nodes is 50% for the considered lattice graph. In our numerical experiments, node states are initialized with values uniformly sampled from the interval $[-1, 1]$. The target states $\boldsymbol{x}^*$ are chosen according to a deterministic cellular automaton rule that produces a pattern in which node states are correlated with the states of their neighbors. The update rule is

$$x_i^{*(n+1)} = 0.95 x_i^{*(n)} + 0.05 \max_{j \in N_i} |A_{ij} x_j^{*(n)}| \,, \tag{E.4}$$

where N_i denotes the neighborhood of node i. For the target state that we use in our simulations, we perform the above iteration for a number of times which is uniformly sampled from the set $\{50, 100, \ldots, 500\}$. For a large number of iterations, the update rule Relation (E.4) creates clusters in which nodes share similar state values. Ultimately, all node states will converge to the initial state value with the highest absolute value.

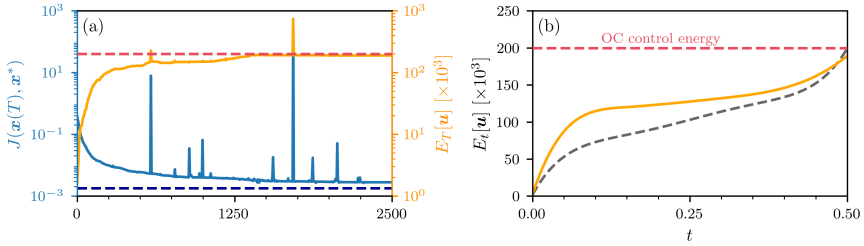


FIGURE E.7: **Control of linear systems with a reduced set of controlled nodes. (a)** Final loss (Relation (6.5) in the main text) and control energy (Relation (2.2) in the main text) as a function of training epochs. Dashed and solid lines indicate optimal control and AI Pontryagin-based solutions. **(b)** Total control energy of AI Pontryagin (solid orange line) and optimal control (dashed grey line). The optimal control solution is based on Relation (6.6) in the main text. The fraction of controlled nodes in (a,b) is 50%. We determined driver matrix B and driver nodes according to the maximum matching method [129].

Figure E.7 shows that AI Pontryagin is able to reach loss and control energy values that are similar to those of optimal control. The neural network that we use in this simulation has 1 hidden layer with 15 ELU hidden units. Bias terms are included as inputs in all layers. Further details about the training process are publicly available at the code repository.

E.5 Regularization Parameter of the Adjoint Gradient Method

The control energy term in the AGM depends on the regularization parameter β. In Figure E.8, we test the performance of the AGM and AI Pontryagin to control coupled Kuramoto oscillators on a square lattice with $N = 225$ nodes and different AGM regularization parameters β ranging from 10^{-7} to 10^{-1}. If the regularization parameter β is too large, the energy term and not the cost associated with a low degree of synchronization dominates, and the AGM fails to synchronize the oscillator system, leading to small order parameters and large control energies.

AI Pontryagin has no explicit energy regularization hyperparameter in the sense of β. It relies on an implicit energy regularization, resulting from the interplay of ANN initialization and an induced gradient descent (see main text). For a more detailed comparison with the AGM, we analyze the dependence of the order parameter and control energy on different learning rates η for AI Pontryagin. We observe that AI Pontryagin achieves stable control of the oscillator system, as indicated by the order parameter value of $r(T) \approx 1$. At the same time, different learning rates may lead to different control energies $E_t(\mathbf{u})$. Small learning rates allow AI Pontryagin to explore

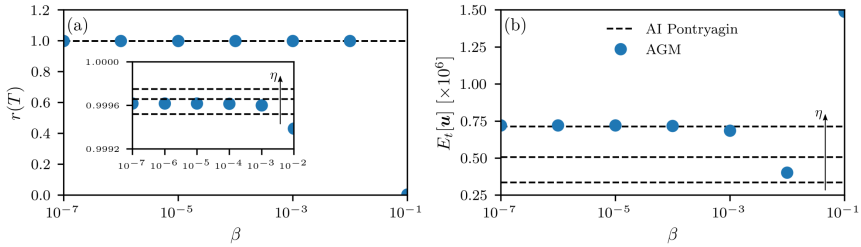


FIGURE E.8: **Controlling coupled oscillators with AI Pontryagin and the AGM.** We test the performance of AI Pontryagin and AGM (Eqs. Relations (6.17) and (6.18)) in the main text] to control coupled Kuramoto oscillators on a square lattice with $N = 225$ nodes and different AGM regularization parameter β ranging from 10^{-7} to 10^{-1}. The total simulation time is $T = 3$. **(a)** The order parameter $r(T)$. **(b)** The control energy $E_t(\mathbf{u})$. Dashed horizontal lines indicate AI Pontryagin (with different learning rates η= 0.22,0.27,0.32) and blue disks indicate AGM solutions ($\tilde{\eta} = 25$).

the dynamical system in more detail, leading to smaller control energies and nearly optimal control solutions.

E.6 Runtime Comparison

In this section, we provide a more detailed comparison of the runtime complexity of AI Pontryagin and the adjoint-gradient method (AGM). To do so, we study the performance of both methods in controlling systems of Kuramoto oscillators whose connections are described by Erdős–Rényi networks. As in the main text, we use a subcritical coupling constant $K = 0.1K^*$ and compare trajectories for which order parameter and control energy values of both methods are similar. Learning rates are reported in Table E.2.

Figure E.9 shows a runtime comparison for 9 different system sizes. We observe that the mean runtime is between 0.5 and 1.5 seconds for AI Pontryagin and it is between 20 and 30 seconds for the AGM. What are the main features in the respective control architectures that are associated with these remarkable 20 to 30-fold speedups of AI Pontryagin? To numerically solve the underlying dynamical systems, we use the Dormand–Prince (DOPRI) method with adaptive step size [333] in both control frameworks. For a network with N nodes, the AGM relies on the numerical integration of $2N$ nonlinear differential equations, describing the coupled primal and adjoint equations (Relations (6.11), (6.12) and (6.18) in the main text) , at every iteration step. After solving the primal and adjoint systems, the AGM calculates a new estimate of the optimal control function according to the gradient descent Relation (6.17) (see main text).

To solve control problems with AI Pontryagin, one needs to numerically integrate the primal system (not the adjoint system) and backpropagate gradients. Differences in the runtime performance between both control frameworks are associated with

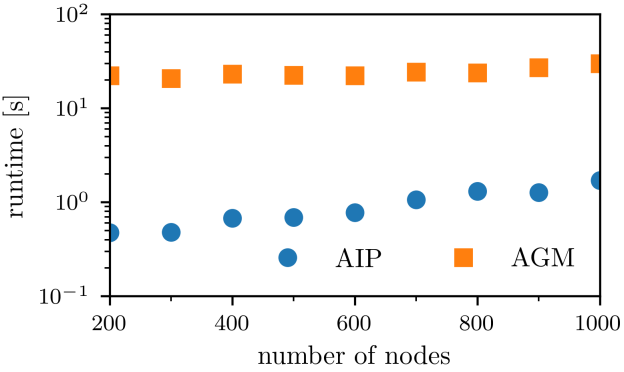


FIGURE E.9: **Runtime comparison between AI Pontryagin and the adjoint-gradient method.** We show the runtime needed to control a network of coupled Kuramoto oscillators with a subcritical coupling constant $K = 0.1K^*$. Blue disks indicate the runtime of AI Pontryagin (AIP) and orange squares indicate the runtime of the adjoint-gradient method (AGM). Simulations were performed on an Erdős–Rényi network $G(N, p)$ with $p = 0.3$. The number of time steps is 60 and the number of oscillators N ranges from 200 to 1,000. Reported runtimes are averaged over 10 realizations. Error bars are smaller than the markers.

stiffness problems that may arise in the numerical solution of the coupled primal and adjoint equations (Relations (6.11), (6.12) and (6.18) in the main text), and the corresponding gradient descent in the control function (Relation (6.17) in the main text). To identify the main computational bottlenecks in the AGM, we performed a detailed runtime analysis of all code segments and found that the adjoint system solver requires very small step sizes to resolve the interaction between the adjoint system and the gradient descent in the control functions. One possibility to further improve the performance of the AGM is to use a problem-tailored initialization of the control function. Such an approach requires knowledge on the mathematical structure of the optimal control signal which may not be possible to obtain, in particular for high-dimensional and analytically intractable control problems. In our simulations, we use a uniform random initialization with support $[1, 2]$ to align the runtimes with those reported in [150]. We also tested other initialization protocols (*e.g.*, $u^{(0)}(t) = 1$ and $u^{(0)}(t) = t/T$) and were able to reduce the runtime for an Erdős–Rényi network with 225 nodes by a factor between 2–3. Still, for this example, AI Pontryagin is about 5–10 times faster, without optimizing the initial guess $u^{(0)}(t)$.

Edges (undirected)	Nodes	Learning rate η	Learning rate $\tilde{\eta}$
5,968	200	0.05	0.6
13,464	300	0.05	0.6
23,865	400	0.05	0.6
37,422	500	0.05	0.6
53,897	600	0.05	0.65
73,444	700	0.05	0.65
96,034	800	0.05	0.6
121,320	900	0.05	0.6
149,938	1,000	0.05	0.6

TABLE E.2: Learning rates Kuramoto dynamics on Erdős–Rényi networks η (AI Pontryagin) and $\tilde{\eta}$ (AGM) used for learning the control of Kuramoto dynamics on Erdős–Rényi networks with different numbers of nodes. All ANNs use stochastic gradient descent for learning and only differ in their learning rate. The number of hidden layers and hidden layer neurons are 1 and 2, respectively. We use an ELU activation function and train the ANN for three epochs. At each node, we include a bias term and set all weights initially to a value of 10^{-3}. The energy regularization parameter of the AGM is set to $\beta = 10^{-7}$.

E.7 Nomenclature

Table E.3: Nomenclature Part I for Chapter 6.

$G(\mathrm{V}, \mathrm{E})$	A graph represented as an ordered pair of a set of nodes V and a set of edges E.
N	The number of nodes in a graph $N = \lvert\mathrm{V}\rvert$.
$\boldsymbol{x}(t)$	A vector $\boldsymbol{x}(t) \in \mathbb{R}^N$, which denotes the state of a dynamical system at time t.
$\boldsymbol{x}^*$	A vector that denotes the target state of a dynamical system.
T	The terminal time for control of a dynamical system.
$\hat{\boldsymbol{u}}(t)$	A control signal value calculated from AI Pontryagin.
$f(\boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}(t)))$	The system evolution function that denotes the dynamic interactions between nodes and drivers when calculating the state derivative.
J	Cost function.
$E_T[\boldsymbol{u}]$	Control energy.
$\boldsymbol{w}$	Weight vector.
(n)	Gradient descent step index.
H	Control Hamiltonian.
λ	The adjoint variable.
$W(T)$	The controllability Gramian matrix.
$\boldsymbol{v}(T)$	Difference between target state and the reached state under no control.
$\boldsymbol{x}^{\mathrm{T}}$	Transpose.
p	p-value.
$\mathcal{J}_{\hat{u}}$	The Jaccobian matrix with elements $\partial \hat{\boldsymbol{u}}_i / \partial \boldsymbol{w}_j$

Table E.5: Nomenclature Part II (Coupled Oscillators) for Chapter 6.

ω_i	Natural frequency for oscillator (node) i.
Ω	Natural frequency vector.
K	Coupling constant
θ_i	phase of oscillator i.
Θ	phase vector.
$\boldsymbol{D}$	the graph degree diagonal matrix $\boldsymbol{D}$ of G, where all off-diagonal elements are 0 and diagonal elements are equal to the degree $D\boldsymbol{D}_{i,i} = d_i$ of the corresponding node.
$\boldsymbol{L}$	the graph Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - A$ of G.
$\boldsymbol{L}^\dagger$	Pseudo-inverse of the graph Laplacian matrix $\boldsymbol{L}$ of G.
$r(t)$	Order parameter, which measures the synchronization of coupled oscillators.
K	Coupling constant.
K^*	Coupling constant, where the system synchronizes without control.
β	Energy regularization constant.
η	Learning rate.
$\widetilde{\eta}$	Learning rate for adjoint gradient method.

Appendix

F

Abbreviations

Table F.1: Abbreviated terms and description for models and methods in general.

AI	Artificial Intelligence.
Ref.	Reference (used when referring to explict work in literature.
SDGs	Sustainable Development Goals.
ICT	Information and Communications Technology.
GPS	Global Positioning System.
FAQ	Frequently Asked Questions.
GPU	Graphic Processing Unit.
CPU	Central Processing Unit.
CO_2	Carbon Dioxide.
SO_2	Sulfur Dioxide.
$PO_4{}^{3-}$	Phosphate
L	liters
g	grams
eq.	equivalent.
m	meters.

Table F.3: Abbreviated terms and descriptions for Organizations, Institutes etc.

UN	United Nations.
WWF	World Wildlife Fund.
VKI	Verein für Konsumenten Information.
EU	European Union.
FAO	Food and Agriculture Organization of the United Nations.

Table F.5: Abbreviated terms and descriptions for neural networks and deep learning.

ANN	Artificial Neural Network.
GNN	Graph Neural Network.
NODEC	Neural Ordinary Differential Equation Control.
NN	Neural Network.
RL	Reinforcement Learning.
TD3	Twin Delayed Deep Deterministic Policy Gradient.
SAC	Soft Actor Critic.
A2C	Asynchronous Actor Critic.
GeLU	Gaussian Error Linear Unit.
ELU	Exponential Linear Unit
MSE	Mean Squarred Error.
ReLU	Rectifier Linear Unit.
DOPRI	Dormand–Prince.

Table F.7: Abbreviated terms related to mathematical modelling and statistics.

AGM	Adjoint Gradient Method.
RND	Random.
TCC	Targeted Constant Control.
OC	Optimal Control.
FC	Feedback Control.
DTW	Dynamic Time Wrapping.
PID	Proportional-Integral-Derivative.
G3A	Gradient Guided Genetic Algorithm.
MO-NES	Multi-Objective Natural Evolution Strategies.
NSGA-II	Non-dominated Sorting Genetic Algorithm II.
RNSGA-II	Reference point Non-dominated Sorting Genetic Algorithm II.
IAM	Integrated Assessment Model.
DICE	Dynamic Integrated Climate Change Model.
ODE	Ordinary Differential Equation.
LTI	Linear Time-Invariant.
PMP	Pontryagin's Maximum Principle.
HJB	Hamilton–Jacobi–Bellman.
SIR	Susceptible Infected Recovered
CDF	Cumulative Distribution/Density Function.
PDF	Probability Distribution/Density Function.
IQR	InterQuantile Range.
CI	Confidence Interval.

Bibliography

[1] Garrett Hardin. "The Tragedy of the Commons". In: *Science* 162.3859 (1968), pp. 1243–1248. ISSN: 0036-8075. DOI: 10.1126/science.162.3859.1243.

[2] Robyn M. Dawes. *Formal Models of Dilemmas in Social Decision-Making*. Vol. 14. 12. Defense Technical Information Center, 1975, pp. 87–107. ISBN: 9780123972507. DOI: 10.1016/B978-0-12-397250-7.50010-6.

[3] Clifford J. Shultz and Morris B Holbrook. "Marketing and the Tragedy of the Commons: A Synthesis, Commentary, and Analysis for Action". In: *Journal of Public Policy & Marketing* 18.2 (1999), pp. 218–229. DOI: 10.2307/30000542.

[4] REN21. *Renewables 2016. Global Status Report-Key Findings*. Tech. rep. REN21, 2016, p. 32.

[5] Jerry M Melillo et al. *Climate change impacts in the United States: the third national climate assessment*. Tech. rep. November 2013. U.S. Global Change Research Program, 2014, p. 841. DOI: 10.7930/j0z31WJ2.

[6] Ning Lin. "Tropical cyclones and heatwaves". In: *Nature Climate Change* 9.8 (2019), pp. 579–580.

[7] Michael Goss et al. "Climate change is increasing the likelihood of extreme autumn wildfire conditions across California". In: *Environmental Research Letters* 15.9 (2020), p. 094016.

[8] Jean-luc Dupuy et al. "Climate change impact on future wildfire danger and activity in southern Europe: a review". In: *Annals of Forest Science* 77.2 (2020), pp. 1–24.

[9] Stephan Harrison et al. "Climate change and the global pattern of moraine-dammed glacial lake outburst floods". In: *The Cryosphere* 12.4 (2018), pp. 1195–1209.

[10] Jiwen Fan et al. "Substantial contribution of anthropogenic air pollution to catastrophic floods in Southwest China". In: *Geophysical Research Letters* 42.14 (2015), pp. 6066–6075.

[11] Joseph Poore and Thomas Nemecek. "Reducing food's environmental impacts through producers and consumers". In: *Science* 360.6392 (2018), pp. 987–992.

[12] SR Veron, JM Paruelo, and M Oesterheld. "Assessing desertification". In: *Journal of Arid Environments* 66.4 (2006), pp. 751–763.

[13] Anthony Oliver-Smith. "Debating environmental migration: society, nature and population displacement in climate change". In: *Journal of International Development* 24.8 (2012), pp. 1058–1070.

[14] R. W. Kates et al. "Sustainability Science". In: *Science* 292.5517 (2001), pp. 641–642. ISSN: 00368075. DOI: 10.1126/science.1059386.

[15] Yossi Maaravi et al. ""The tragedy of the commons": How individualism and collectivism affected the spread of the COVID-19 pandemic". In: *Frontiers in Public Health* 9 (2021), p. 37.

[16] Dirk Helbing and Evangelos Pournaras. "Build digital democracy A WISE KING?" In: *Nature* 527.7576 (2015), pp. 33–34. ISSN: 0028-0836. DOI: 10.1038/527033a.

[17] Laura Carmichael, Sophie Stalla-Bourdillon, and Steffen Staab. "Data Mining and Automated Discrimination: A Mixed Legal/Technical Perspective". In: *IEEE Intelligent Systems* 31.6 (2016), pp. 51–55. ISSN: 1541-1672. DOI: 10.1109/MIS.2016.96.

[18] B Liscouski and W Elliot. "Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations". In: *A report to US Department of Energy* 40.4 (2004), p. 86.

[19] Johannes Klinglmayr, Bernhard Bergmair, and Evangelos Pournaras. "Sustainable Consumerism via Self-Regulation". In: *2016 IEEE 1st International Workshops on Foundations and Applications of Self*Systems (FAS*)*. IEEE, 2016, pp. 138–142. ISBN: 978-1-5090-3651-6. DOI: 10.1109/FAS-W.2016.40.

[20] UN. *Global Sustainable Development Report: 2015 edition*. Tech. rep. United Nations, 2015, p. 202.

[21] C H Douglas. *Sustainable development in European Union states small island dependencies–strategies and targets*. Tech. rep. Eurostat, 1997, pp. 181–186. DOI: 10.1002/(SICI)1099-0976(199711)7:6<181::AID-EET134>3.0.CO;2-S.

[22] William Colglazier. "Sustainable development agenda: 2030." In: *Science* 349.6252 (2015), pp. 1048–1050. ISSN: 00368075. DOI: 10.1126/science.aad2333.

[23] Ricardo Vinuesa et al. "The role of artificial intelligence in achieving the Sustainable Development Goals". In: *Nature communications* 11.1 (2020), pp. 1–10.

[24] United Nations. *United Nations Sustainable Development Goals*. 2016. URL: https://www.un.org/sustainabledevelopment/news/communications-material/ (visited on 08/03/2021).

[25] Zhiqiang Zhao et al. "Synergies and tradeoffs among Sustainable Development Goals across boundaries in a metacoupled world". In: *Science of the Total Environment* 751 (2021), p. 141749.

[26] Rohan Doshi et al. “Maximizing sustainability of ecosystem model through socio-economic policies derived from multivariable optimal control theory”. In: *Clean Technologies and Environmental Policy* 17.6 (2015), pp. 1573–1583.

[27] W.A. Brock, K.G. Mäler, and C. Perrings. *Resilience and Sustainability: The Economic Analysis of Non-linear Dynamic Systems*. Beijer discussion paper series. Beijer International Institute of Ecological Economics, 2000.

[28] Michael D Mastrandrea and Stephen H Schneider. “Integrated assessment of abrupt climatic changes”. In: *Climate Policy* 1.4 (2001), pp. 433–449.

[29] William D Nordhaus. “Optimal greenhouse-gas reductions and tax policy in the" DICE" model”. In: *The American Economic Review* 83.2 (1993), pp. 313–317.

[30] Pahola T Benavides, Urmila Diwekar, and Heriberto Cabezas. “Controllability of complex networks for sustainable system dynamics”. In: *Journal of Complex Networks* 3.4 (2015), pp. 566–583.

[31] Kirk Hamilton, Kirk Hamilton, and Giles Atkinson. *Wealth, welfare and sustainability: Advances in measuring sustainable development*. Edward Elgar Publishing, 2006.

[32] Nauman Sheikh. *Implementing analytics: A blueprint for design, development, and adoption*. Newnes, 2013.

[33] Lennart Sjöberg. “Intuitive vs. analytical decision making: which is preferred?” In: *Scandinavian Journal of Management* 19.1 (2003), pp. 17–29.

[34] Roberto Poli. *Handbook of anticipation: Theoretical and applied aspects of the use of future in decision making*. Springer, 2019.

[35] Andrew M Pettigrew. *The politics of organizational decision-making*. Routledge, 2014.

[36] Gerd Gigerenzer and Wolfgang Gaissmaier. “Decision making: Nonrational theories”. In: *International encyclopedia of the social & behavioral sciences*. Elsevier, 2015, pp. 911–916.

[37] Davor Antanasijević et al. “The forecasting of municipal waste generation using artificial neural networks and sustainability indicators”. In: *Sustainability science* 8.1 (2013), pp. 37–46.

[38] Seyed Mohsen Sadatiyan Abkenar et al. “Evaluation of genetic algorithms using discrete and continuous methods for pump optimization of water distribution systems”. In: *Sustainable Computing: Informatics and Systems* 8 (2015), pp. 18–23.

[39] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific, 2012.

[40] Sabina Tomkins et al. “Sustainability at scale: towards bridging the intention-behavior gap with sustainable recommendations”. In: *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM. 2018, pp. 214–218.

[41] William D Nordhaus et al. *Thinking about carbon dioxide: theoretical and empirical aspects of optimal control strategies*. Tech. rep. Cowles Foundation for Research in Economics, Yale University, 1980.

[42] Edward A Parson, Fisher-Vanden, and Karen. "Integrated assessment models of global climate change". In: *Annual Review of Energy and the Environment* 22.1 (1997), pp. 589–628.

[43] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2017, p. 775. ISBN: 9780262035613.

[44] Nicola Guarino and Pierdaniele Giaretta. "Ontologies and knowledge bases". In: *Towards very large knowledge bases* (1995), pp. 1–2.

[45] Thomas Schmid. "Using Learning Algorithms to Create, Exploit and Maintain Knowledge Bases: Principles of Constructivist Machine Learning." In: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1)*. 2020.

[46] Michael Sharp, Ronay Ak, and Thomas Hedberg Jr. "A survey of the advancing use and development of machine learning in smart manufacturing". In: *Journal of manufacturing systems* 48 (2018), pp. 170–179.

[47] Erik Brynjolfsson and Tom Mitchell. "What can machine learning do? Workforce implications". In: *Science* 358.6370 (2017), pp. 1530–1534.

[48] Costanza Conforti et al. "Natural Language Processing for Achieving Sustainable Development: The Case of Neural Labelling to Enhance Community Profiling". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 8427–8444.

[49] Thomas Stark. "Using deep convolutional neural networks for the identification of informal settlements to improve a sustainable development in urban environments". PhD thesis. Technische Universität München, 2018.

[50] Ying Lien Lee and Fei Hui Huang. "Recommender system architecture for adaptive green marketing". In: *Expert Systems with Applications* 38.8 (2011), pp. 9696–9703. ISSN: 09574174. DOI: 10.1016/j.eswa.2011.01.164.

[51] Virginia Dignum. *Responsible artificial intelligence: how to develop and use AI in a responsible way*. Springer Nature, 2019.

[52] Efthimios Bothos, Dimitris Apostolou, and Gregoris Mentzas. "Recommender systems for nudging commuters towards eco-friendly decisions". In: *Intelligent Decision Technologies* 9.3 (2015), pp. 295–306.

[53] Viktoria Spaiser et al. "The sustainable development oxymoron: quantifying and modelling the incompatibility of sustainable development goals". In: *International Journal of Sustainable Development & World Ecology* 24.6 (2017), pp. 457–470.

[54] Stuart E. Middleton, David De Roure, and Nigel R. Shadbolt. "Ontology-Based Recommender Systems". In: *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 779–796. DOI: 10.1007/978-3-540-92673-3_35.

[55] Mario Rodriguez, Christian Posse, and Ethan Zhang. "Multiple objective optimization in recommender systems". In: *Proceedings of the sixth ACM conference on Recommender systems*. 2012, pp. 11–18.

[56] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. "Collaborative deep learning for recommender systems". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1235–1244.

[57] Li Chen et al. "Human decision making and recommender systems". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 3.3 (2013), pp. 1–7.

[58] Da Ruan, Frank Hardeman, and Klaas van der Meer. *Intelligent decision and policy making support systems*. Vol. 117. Springer, 2008.

[59] Long-Sheng Chen et al. "Developing recommender systems with the consideration of product profitability for sellers". In: *Information Sciences* 178.4 (2008), pp. 1032–1048.

[60] Dirk Bergemann and Deran Ozmen. "Optimal pricing with recommender systems". In: *Proceedings of the 7th ACM Conference on Electronic Commerce*. 2006, pp. 43–51.

[61] J Ben Schafer, Joseph Konstan, and John Riedl. "Recommender systems in e-commerce". In: *Proceedings of the 1st ACM conference on Electronic commerce*. 1999, pp. 158–166.

[62] Xue-Wen Chen and Xiaotong Lin. "Big data deep learning: challenges and perspectives". In: *IEEE access* 2 (2014), pp. 514–525.

[63] Luke Taylor and Geoff Nitschke. "Improving deep learning with generic data augmentation". In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2018, pp. 1542–1547.

[64] Qingsong Wen et al. "Time Series Data Augmentation for Deep Learning: A Survey". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Ed. by Zhi-Hua Zhou. ijcai.org, 2021, pp. 4653–4660. DOI: 10.24963/ijcai.2021/631.

[65] R. Kitchin. *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*. SAGE Publications, 2014. ISBN: 9781473908260.

[66] S. Gkika and G. Lekakos. "Investigating the Effectiveness of Persuasion Strategies on Recommender Systems". In: *2014 9th International Workshop on Semantic and Social Media Adaptation and Personalization*. 2014, pp. 94–97. DOI: 10.1109/SMAP.2014.37.

[67] Daniel L Rosenfeld, Hank Rothgerber, and A Janet Tomiyama. "From mostly vegetarian to fully vegetarian: Meat avoidance and the expression of social identity". In: *Food Quality and Preference* 85 (2020), p. 103963.

[68] Federico Cugurullo. "Urban artificial intelligence: From automation to autonomy in the smart city". In: *Frontiers in Sustainable Cities* 2 (2020), p. 38.

[69] JinHyo Joseph Yun et al. "Not deep learning but autonomous learning of open innovation for sustainable artificial intelligence". In: *Sustainability* 8.8 (2016), p. 797.

[70] Evangelos Pournaras, Peter Pilgerstorfer, and Thomas Asikis. "Decentralized collective learning for self-managed sharing economies". In: *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 13.2 (2018), pp. 1–33.

[71] Batya Friedman. "User Autonomy". In: *ACM SIGCHI Bulletin* 30.1 (1998), pp. 26–29. ISSN: 07366906. DOI: 10.1145/280571.280583.

[72] Dirk Helbing. *Thinking ahead-essays on big data, digital revolution, and participatory market society*. Springer, 2015, pp. 1–194. ISBN: 9783319150789. DOI: 10.1007/978-3-319-15078-9. arXiv: arXiv:1011.1669v3.

[73] Jeroen Van den Hoven. "ICT and value sensitive design". In: *The information society: Innovation, legitimacy, ethics and democracy in honor of Professor Jacques Berleur SJ*. Springer, 2007, pp. 67–72.

[74] Margaret A Goralski and Tay Keong Tan. "Artificial intelligence and sustainable development". In: *The International Journal of Management Education* 18.1 (2020), p. 100330.

[75] Jan Zabojnik. "Centralized and decentralized decision making in organizations". In: *Journal of Labor Economics* 20.1 (2002), pp. 1–22.

[76] John Nikolas Tsitsiklis. *Problems in decentralized decision making and computation*. Tech. rep. Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

[77] Per Sebastian Skardal and Alex Arenas. "Control of coupled oscillator networks with application to microgrid technologies". In: *Science Advances* 1.7 (2015).

[78] Cui Jianguo and Wang Junling. "Study on Optimal Control Models of Water Supply System in City". In: *JOURNAL-TAIYUAN UNIVERSITY OF TECHNOLOGY* 33.3 (2002), pp. 285–289.

[79] Pearl Pu and Li Chen. "A User - Centric Evaluation Framework for Recommender Systems". In: *Proceedings of the 5th ACM conference on Recommender systems - RecSys '11* (2011), pp. 157–164. ISSN: 1450306837. DOI: 10.1145/2043932.2043962.

[80] S M McNee, J Riedl, and J a Konstan. "Being accurate is not enough: how accuracy metrics have hurt recommender systems". In: *CHI'06 extended abstracts on Human factors in computing systems*. New York, New York, USA: ACM Press, 2006, p. 1101. ISBN: 1595932984. DOI: 10.1145/1125451.1125659.

[81] John O'Donovan and Barry Smyth. "Trust in recommender systems". In: *Proceedings of the 10th international conference on Intelligent user interfaces IUI 05* 05pages.June (2005), p. 167. ISSN: 1875-7650. DOI: 10.1145/1040830.1040870.

[82] Paolo Massa and Paolo Avesani. "Trust Metrics in Recommender Systems". In: *Computing with Social Trust, Human-Computer Interaction Series* (2009), pp. 259–285. ISSN: 1571-5035. DOI: 10.1007/978-1-84800-356-9.

[83] Florian Dorfler and Francesco Bullo. "Synchronization and transient stability in power networks and nonuniform Kuramoto oscillators". In: *SIAM Journal on Control and Optimization* 50.3 (2012), pp. 1616–1642.

[84] Benjamin F Maier and Dirk Brockmann. "Effective containment explains subexponential growth in recent confirmed COVID-19 cases in China". In: *Science* 368.6492 (2020), pp. 742–746.

[85] Berthold Göttgens. "Regulatory network control of blood stem cells". In: *Blood* 125.17 (2015), pp. 2614–2620.

[86] Danilo Delpini et al. "Evolution of controllability in interbank networks". In: *Scientific Reports* 3.1 (2013), pp. 1–5.

[87] Benjamin Schäfer et al. "Dynamically induced cascading failures in power grids". In: *Nature Communications* 9.1 (2018), pp. 1–13.

[88] Wongyeong Choi and Eunha Shim. "Optimal strategies for social distancing and testing to control COVID-19". In: *Journal of Theoretical Biology* 512 (2021), p. 110568.

[89] Kay H Brodersen et al. "Inferring causal impact using Bayesian structural time-series models". In: *The Annals of Applied Statistics* 9.1 (2015), pp. 247–274.

[90] Thomas Asikis et al. "How value-sensitive design can empower sustainable consumption". In: *Royal Society open science* 8.1 (2021), p. 201418.

[91] *Dunnhumby - The complete journey dataset*. Available at `https://www.dunnhumby.com/source-files/` (last accessed: September 2021). 2021.

[92] *FAO – Food Agriculture Organization, Food Balance Sheets*. Available at `http://www.fao.org/3/X9892E/X9892e05.htm` (last accessed: September 2021). 20211.

[93] Paul Voigt and Axel Von dem Bussche. Vol. 10. Springer, 2017. ISBN: 978-3-319-86291-0. DOI: 10.1007/978-3-319-57959-7.

[94] Cynthia Dwork. "Differential Privacy: A Survey of Results". In: *Theory and Applications of Models of Computation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19. DOI: 10.1007/978-3-540-79228-4_1.

[95] Irish Social Science Data Archive., ed. *CER Smart Metering Project - Electricity Customer Behaviour Trial*. 2010. URL: https://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[96] Thomas Asikis and Evangelos Pournaras. "Optimization of privacy-utility trade-offs under informational self-determination". In: *Future Generation Computer Systems* 109 (2020), pp. 488–499.

[97] Julien Perolat et al. "A multi-agent reinforcement learning model of common-pool resource appropriation". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 3646–3655.

[98] Anna Nagurney. *Supply chain network economics: dynamics of prices, flows and profits*. Edward Elgar Publishing, 2006.

[99] Zhengfu Rao and Elad Salomons. "Development of a real-time, near-optimal control process for water-distribution networks". In: *Journal of Hydroinformatics* 9.1 (2007), pp. 25–37.

[100] Talha Manzoor et al. "Optimal control for sustainable consumption of natural resources". In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 10725–10730.

[101] Jerzy Skrzypczyk. "Accuracy analysis of statistical linearization methods applied to nonlinear dynamical systems". In: *Reports on mathematical physics* 36.1 (1995), pp. 1–20.

[102] Christian P Traeger. "A 4-stated DICE: quantitatively addressing uncertainty effects in climate change". In: *Environmental and Resource Economics* 59.1 (2014), pp. 1–37.

[103] Z Kowalczuk. "Discrete approximation of continuous-time systems: a survey". In: *IEE Proceedings G (Circuits, Devices and Systems)* 140.4 (1993), pp. 264–278.

[104] Richard SJ Tol. "On the optimal control of carbon dioxide emissions: an application of FUND". In: *Environmental Modeling & Assessment* 2.3 (1997), pp. 151–163.

[105] Dima Nazzal et al. "Product servicing for lifespan extension and sustainable consumption: An optimization approach". In: *International Journal of Production Economics* 142.1 (2013), pp. 105–114.

[106] Sung Jin Yoo, Jin Bae Park, and Yoon Ho Choi. "Stable predictive control of chaotic systems using self-recurrent wavelet neural network". In: *International Journal of Control, Automation and Systems* 3.1 (2005), pp. 43–55.

[107] Philipp Holl, Vladlen Koltun, and Nils Thuerey. "Learning to control pdes with differentiable physics". In: *International Conference on Learning Representations*. 2020.

[108] Brandon Amos et al. "Differentiable MPC for end-to-end planning and control". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018, pp. 8289–8300.

[109] Lucas Böttcher, Hans J Herrmann, and Hans Gersbach. "Clout, activists and budget The road to presidency". In: *PLOS ONE* 13.3 (2018).

[110] Moritz Hoferer et al. "The impact of technologies in political campaigns". In: *Physica A: Statistical Mechanics and its Applications* 538 (2020), p. 122795.

[111] Dirk Brockmann and Dirk Helbing. "The hidden geometry of complex, network-driven contagion phenomena". In: *Science* 342.6164 (2013), pp. 1337–1342.

[112] Nino Antulov-Fantulin et al. “Identification of patient zero in static and temporal networks: Robustness and limitations”. In: *Physical Review Letters* 114.24 (2015), p. 248701.

[113] Lucas Böttcher, JS Andrade, and Hans J Herrmann. “Targeted Recovery as an Effective Strategy against Epidemic Spreading”. In: *Scientific Reports* 7.1 (2017), pp. 1–7.

[114] Francisco A Rodrigues et al. “The Kuramoto model in complex networks”. In: *Physics Reports* 610 (2016), pp. 1–98.

[115] MK Stephen Yeung and Steven H Strogatz. “Time delay in the Kuramoto model of coupled oscillators”. In: *Physical Review Letters* 82.3 (1999), p. 648.

[116] Danilo Delpini et al. “Evolution of Controllability in Interbank Networks”. In: *Scientific Reports* 3.1 (Apr. 2013). DOI: 10.1038/srep01626.

[117] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*. Vol. 1. Elsevier, 1992.

[118] J. E. Moyal. “Stochastic Processes and Statistical Physics”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 11.2 (1949), pp. 150–210. ISSN: 00359246.

[119] Hakan Andersson and Tom Britton. *Stochastic epidemic models and their statistical analysis*. Vol. 151. Springer Science & Business Media, 2012.

[120] Claudio Castellano and Romualdo Pastor-Satorras. “Thresholds for epidemic spreading in networks”. In: *Physical Review Letters* 105.21 (2010), p. 218701.

[121] James P Gleeson. “Binary-state dynamics on complex networks: Pair approximation and beyond”. In: *Physical Review X* 3.2 (2013), p. 021004.

[122] Baruch Barzel and Albert-László Barabási. “Universality in network dynamics”. In: *Nature Physics* 9.10 (2013), pp. 673–681.

[123] Duncan J Watts and Steven H Strogatz. “Collective dynamics of ‘small-world’networks”. In: *Nature* 393.6684 (1998), p. 440.

[124] Derek J De Solla Price. “Networks of scientific papers”. In: *Science* (1965), pp. 510–515.

[125] Albert-László Barabási and Réka Albert. “Emergence of scaling in random networks”. In: *Science* 286.5439 (1999), pp. 509–512.

[126] Michelle Girvan and Mark EJ Newman. “Community structure in social and biological networks”. In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826.

[127] Sergey N Dorogovtsev, Alexander V Goltsev, and José FF Mendes. “Critical phenomena in complex networks”. In: *Reviews of Modern Physics* 80.4 (2008), p. 1275.

[128] Jure Leskovec et al. “Kronecker graphs: An approach to modeling networks”. In: *Journal of Machine Learning Research* 11.Feb (2010), pp. 985–1042.

[129] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. "Controllability of complex networks". In: *Nature* 473.7346 (2011), pp. 167–173.

[130] Yang-Yu Liu and Albert-László Barabási. "Control principles of complex systems". In: *Reviews of Modern Physics* 88.3 (2016), p. 035006.

[131] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[132] Romualdo Pastor-Satorras et al. "Epidemic processes in complex networks". In: *Reviews of Modern Physics* 87 (3 Aug. 2015), pp. 925–979. DOI: 10.1103/RevModPhys.87.925.

[133] Olvi L Mangasarian. "Sufficient conditions for the optimal control of nonlinear systems". In: *SIAM Journal on Control* 4.1 (1966), pp. 139–152.

[134] Morton I Kamien and Nancy L Schwartz. "Sufficient conditions in optimal control theory". In: *Journal of Economic Theory* 3.2 (1971), pp. 207–214.

[135] EJ McShane. "The calculus of variations from the beginning through optimal control theory". In: *SIAM Journal on Control and Optimization* 27.5 (1989), pp. 916–939.

[136] Ricky T. Q. Chen et al. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems* (2018).

[137] Rudolf Emil Kalman et al. "Contributions to the theory of optimal control". In: *Boletín de la Sociedad Matemática Mexicana* 5.2 (1960), pp. 102–119.

[138] Malo LJ Hautus. "Controllability and observability conditions of linear autonomous systems". In: *Indagationes Mathematicae (Proceedings)*. Vol. 72. 1969, pp. 443–448.

[139] Ching-Tai Lin. "Structural controllability". In: *IEEE Transactions on Automatic Control* 19.3 (1974), pp. 201–208.

[140] Justin Ruths and Derek Ruths. "Control profiles of complex networks". In: *Science* 343.6177 (2014), pp. 1373–1376.

[141] Gang Yan et al. "Controlling complex networks: How much energy is needed?" In: *Physical Review Letters* 108.21 (2012), p. 218703.

[142] XY Zhou. "Maximum principle, dynamic programming, and their connection in deterministic control". In: *Journal of Optimization Theory and Applications* 65.2 (1990), pp. 363–373.

[143] Wendell H Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*. Vol. 25. Springer Science & Business Media, 2006.

[144] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Princeton university press, 2015.

[145] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[146] Bahare Kiumarsi et al. “Optimal and autonomous control using reinforcement learning: A survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 29.6 (2017), pp. 2042–2062.

[147] Halina Frankowska. “Nonsmooth solutions of Hamilton-Jacobi-Bellman equation”. In: *Modeling and Control of Systems*. Springer-Verlag, 1989, pp. 131–147. DOI: 10.1007/bfb0041191.

[148] FW Lewis, Suresh Jagannathan, and Aydin Yesildirak. *Neural network control of robot manipulators and non-linear systems*. CRC press, 2020.

[149] Bo Pang, Zhong-Ping Jiang, and Iven Mareels. “Reinforcement learning for adaptive optimal control of continuous-time linear periodic systems”. In: *Automatica* 118 (2020), p. 109035.

[150] Umberto Biccari and Enrique Zuazua. “A Stochastic Approach to the Synchronization of Coupled Oscillators”. In: *Frontiers in Energy Research* 8.115 (2020), p. 115. ISSN: 2296-598X. DOI: 10.3389/fenrg.2020.00115.

[151] Arie Wahyu Wijayanto and Tsuyoshi Murata. “Flow-aware vertex protection strategy on large social networks”. In: *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 2017, pp. 58–63.

[152] Sean P Cornelius, William L Kath, and Adilson E Motter. “Realistic control of network dynamics”. In: *Nature Communications* 4.1 (2013), pp. 1–9.

[153] Per Sebastian Skardal and Alex Arenas. “On controlling networks of limit-cycle oscillators”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26.9 (2016).

[154] Tuomas Haarnoja et al. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1861–1870. URL: https://proceedings.mlr.press/v80/haarnoja18b.html.

[155] Scott Fujimoto, Herke Hoof, and David Meger. “Addressing Function Approximation Error in Actor-Critic Methods”. In: *International Conference on Machine Learning*. 2018, pp. 1587–1596.

[156] Mark Newman. *Networks*. Oxford: Oxford university press, 2018.

[157] Otto Mayr. “The origins of feedback control”. In: *Scientific American* 223.4 (1970), pp. 110–119.

[158] Lawrence F Shampine. *Numerical solution of ordinary differential equations*. Routledge, 2018.

[159] Christian Commault, Jean-Michel Dion, and Jacob W van der Woude. “Characterization of generic properties of linear structured systems for efficient computations”. In: *Kybernetika* 38.5 (2002), pp. 503–520.

[160] Takeo Yamada and Leslie R Foulds. "A graph-theoretic approach to investigate structural and qualitative properties of systems: A survey". In: *Networks* 20.4 (1990), pp. 427–452.

[161] Michael R Garey and David S Johnson. *Computers and intractability*. Vol. 174. Freeman San Francisco, 1979.

[162] Alex Olshevsky. "Minimal controllability problems". In: *IEEE Transactions on Control of Network Systems* 1.3 (2014), pp. 249–258.

[163] Atılım Günes Baydin et al. "Automatic differentiation in machine learning: a survey". In: *Journal of Machine Learning Research* 18.1 (2017), pp. 5595–5637.

[164] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8026–8037.

[165] Patrick Kidger et al. "Neural controlled differential equations for irregular time series". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

[166] Marshall H Stone. "The generalized Weierstrass approximation theorem". In: *Mathematics Magazine* 21.5 (1948), pp. 237–254.

[167] Ding-Xuan Zhou. "Universality of deep convolutional neural networks". In: *Applied and Computational Harmonic Analysis* 48.2 (2020), pp. 787–794.

[168] Anton Maximilian Schäfer and Hans Georg Zimmermann. "Recurrent neural networks are universal approximators". In: *International Conference on Artificial Neural Networks*. Springer. 2006, pp. 632–640.

[169] Eduardo D Sontag and H Siegelmann. "On the computational power of neural nets". In: *Journal of Computer and System Sciences* 50 (1995), pp. 132–150.

[170] Boris Hanin and Mark Sellke. "Approximating continuous functions by relu nets of minimal width". In: *arXiv preprint arXiv:1710.11278* (2017).

[171] John Lygeros. "On reachability and minimum cost optimal control". In: *Automatica* 40.6 (2004), pp. 917–927.

[172] Takeshi Teshima et al. "Universal approximation property of neural ordinary differential equations". In: *arXiv preprint arXiv:2012.02414* (2020).

[173] Michele Alessandro Bucci et al. "Control of chaotic systems by deep reinforcement learning". In: *Proceedings of the Royal Society A* 475.2231 (2019), p. 20190351.

[174] Changchun Hua and Xinping Guan. "Adaptive control for chaotic systems". In: *Chaos, Solitons & Fractals* 22.1 (2004), pp. 55–60.

[175] Shubhendu Bhasin et al. "A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems". In: *Automatica* 49.1 (2013), pp. 82–92.

[176] Bela G Liptak. *Instrument Engineers' Handbook, Volume Two: Process Control and Optimization*. CRC press, 2018.

[177] Alberto Bemporad et al. "The explicit linear quadratic regulator for constrained systems". In: *Automatica* 38.1 (2002), pp. 3–20.

[178] James Sethna et al. *Statistical mechanics: entropy, order parameters, and complexity*. Vol. 14. Oxford University Press, 2006.

[179] Markus Brede. "Locals vs. global synchronization in networks of non-identical Kuramoto oscillators". In: *The European Physical Journal B* 62.1 (2008), pp. 87–94.

[180] Richard S Sutton. "Temporal Credit Assignment in Reinforcement Learning." In: *PhD dissertation* (1985).

[181] Francesco Varoli et al. *Improved Memories Learning*. 2020. arXiv: `2008.10433 [cs.LG]`.

[182] Siddhartha Verma, Guido Novati, and Petros Koumoutsakos. "Efficient collective swimming by harnessing vortices through deep reinforcement learning". In: *Proceedings of the National Academy of Sciences* 115.23 (2018), pp. 5849–5854.

[183] Guido Novati and Petros Koumoutsakos. "Remember and Forget for Experience Replay". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 4851–4860. URL: `https://proceedings.mlr.press/v97/novati19a.html`.

[184] Thomas Asikis, Lucas Böttcher, and Nino Antulov-Fantulin. *Github repository for Neural Network Control*. 2020. URL: `https://github.com/asikist/nnc`.

[185] Thomas Asikis, Lucas Böttcher, and Nino Antulov-Fantulin. *Code Ocean capsule for reproducing all NODEC experiments*. 2021. URL: `https://codeocean.com/capsule/1934600/tree/v1`.

[186] Thomas Asikis, Lucas Böttcher, and Nino Antulov-Fantulin. *Data repository for all NODEC experiments*. 2020. URL: `https://ieee-dataport.org/documents/neural-ordinary-differential-equation-control-dynamics-graphs`.

[187] David Cumin and CP Unsworth. "Generalising the Kuramoto model for the study of neuronal synchronisation in the brain". In: *Physica D: Nonlinear Phenomena* 226.2 (2007), pp. 181–196.

[188] DA Schoenwald and U Ozguner. "Optimal control of feedback linearizable systems". In: *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*. IEEE. 1992, pp. 2033–2034.

[189] Christian Bick, Mark J Panaggio, and Erik A Martens. "Chaos in Kuramoto oscillator networks". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.7 (2018), p. 071102.

[190] Yuri L Maistrenko, Oleksandr V Popovych, and Peter A Tass. "Chaotic attractor in the Kuramoto model". In: *International Journal of Bifurcation and Chaos* 15.11 (2005), pp. 3457–3466.

[191] Yoshiki Kuramoto. "Self-entrainment of a population of coupled non-linear oscillators". In: *International Symposium on Mathematical Problems in Theoretical Physics*. Springer. 1975, pp. 420–422.

[192] Yoshua Bengio et al. "Curriculum learning". In: *International Conference on Machine Learning*. 2009, pp. 41–48.

[193] Karl Johan Åström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton Univeristy Press, 2010.

[194] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *International Conference on Machine Learning*. 2015, pp. 448–456.

[195] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[196] Marcel Salathé and James H Jones. "Dynamics and control of diseases in networks with community structure". In: *PLOS Computational Biology* 6.4 (2010).

[197] Mingtao Xia, Lucas Böttcher, and Tom Chou. *Controlling epidemics through optimal allocation of test kits and vaccine doses across networks*. 2021. arXiv: `2107.13709 [q-bio.PE]`.

[198] Volodymyr Mnih et al. "Asynchronous methods for deep reinforcement learning". In: *International Conference on Machine Learning*. 2016, pp. 1928–1937.

[199] Franco Scarselli et al. "The graph neural network model". In: *IEEE Transactions on Neural Networks* 20.1 (2008), pp. 61–80.

[200] Mandeep Rathee et al. *Learnt Sparsification for Interpretable Graph Neural Networks*. 2021. arXiv: `2106.12920 [cs.LG]`.

[201] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[202] Daron Acemoglu et al. "Optimal targeted lockdowns in a multi-group SIR model". In: *NBER Working Paper* 27102 (2020).

[203] Victor M Preciado et al. "Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks". In: *IEEE Conference on Decision and Control*. IEEE. 2013, pp. 7486–7491.

[204] Jiezhong Qiu et al. "Deepinf: Social influence prediction with deep learning". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2110–2119.

[205] A. Parodi et al. “The potential of future foods for sustainable and healthy diets”. In: *Nature Sustainability* 1.12 (2018), pp. 782–789. ISSN: 2398-9629. DOI: 10.1038/s41893-018-0189-7.

[206] Jeffrey D. Sachs et al. “Six Transformations to achieve the Sustainable Development Goals”. In: *Nature Sustainability* 2.9 (2019), pp. 805–814. ISSN: 2398-9629. DOI: 10.1038/s41893-019-0352-9.

[207] UN Desa et al. “Transforming our world: The 2030 agenda for sustainable development”. In: *United Nations* A/RES/70/1 (2016).

[208] Rob Vos, Andrea Cattaneo, et al. “The state of food and agriculture”. In: *Food and Agricultural Organization of the United Nations* (2017).

[209] Bojana Bajželj, Julian M. Allwood, and Jonathan M. Cullen. “Designing Climate Change Mitigation Plans That Add Up”. In: *Environmental Science & Technology* 47.14 (2013). PMID: 23799265, pp. 8062–8069. DOI: 10.1021/es400399h.

[210] Tim Searchinger et al. *Creating a sustainable food future: A menu of solutions to feed nearly 10 billion people by 2050. Final report.* World Resources Institute, 2019. ISBN: 978-1-56973-963-1.

[211] Tomoko Hasegawa et al. “Tackling food consumption inequality to fight hunger without pressuring the environment”. In: *Nature Sustainability* 2.9 (2019), pp. 826–833.

[212] Francesco Fuso Nerini et al. “Connecting climate action with other sustainable development goals”. In: *Nature Sustainability* 2.8 (2019), pp. 674–680.

[213] Klaus G Grunert, Sophie Hieke, and Josephine Wills. “Sustainability labels on food products: Consumer motivation, understanding and use”. In: *Food Policy* 44 (2014), pp. 177–189.

[214] Ralph E Horne. “Limits to labels: The role of eco-labels in the assessment of product sustainability and routes to sustainable consumption”. In: *International Journal of consumer studies* 33.2 (2009), pp. 175–182.

[215] Hannah Winkler von Mohrenfels and Daniel Klapper. “The influence of mobile product information on brand perception and willingness to pay for green and sustainable products”. In: *International Conference on Information Systems, ICIS 2012*. Vol. 1. 2012, pp. 872–887.

[216] Raja Chatila and John C Havens. “The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems”. In: *Robotics and Well-Being*. Springer, 2019, pp. 11–16.

[217] Sungchul Choi and Alex Ng. “Environmental and economic dimensions of sustainability and price effects on consumer responses”. In: *Journal of business ethics* 104.2 (2011), pp. 269–282.

[218] *ASSET EU Project*. http://www.asset-consumerism.eu. Accessed March 2020. 2020.

[219] *Coop*. Available at https://www.coop.ee (last accessed: March 2020). 2020.

[220] *Winkler Markt*. Available at `http://www.winklermarkt.at` (last accessed: March 2020). 2020.

[221] Arik Friedman et al. "Privacy aspects of recommender systems". In: *Recommender Systems Handbook*. Springer, 2015, pp. 649–688.

[222] Dirk Helbing et al. "Will democracy survive big data and artificial intelligence?" In: *Towards digital enlightenment*. Springer, 2019, pp. 73–98.

[223] Felix Von Reischach et al. "Evaluation of 1D barcode scanning on mobile phones". In: *2010 Internet of Things (IOT)*. IEEE. 2010, pp. 1–5.

[224] Leander B Hormann et al. "Augmented Shopping Experience for Sustainable Consumption Using the Internet of Thing". In: *IEEE Internet of Things Magazine* 2.3 (2019), pp. 46–51.

[225] Nils Ole Tippenhauer et al. "On the requirements for successful GPS spoofing attacks". In: *Proceedings of the 18th ACM conference on Computer and communications security*. ACM. 2011, pp. 75–86.

[226] Barry Smith. "Ontology (science)". In: *Nature Precedings* (2008).

[227] Michael Ashburner et al. "Gene ontology: tool for the unification of biology". In: *Nature genetics* 25.1 (2000), pp. 25–29.

[228] Judith Blake. "Bio-ontologies–fast and furious". In: *Nature biotechnology* 22.6 (2004), pp. 773–774.

[229] Steffen Fritz et al. "Citizen science and the United Nations Sustainable Development Goals". In: *Nature Sustainability* 2.10 (2019), pp. 922–930.

[230] Payam Aminpour et al. "Wisdom of stakeholder crowds in complex social–ecological systems". In: *Nature Sustainability* (2020), pp. 1–9.

[231] Gerrit Stöckigt et al. "Assessment of Consumer Attitudes Toward Sustainability in Food Logistics and the Role of Shopping Behavior and Personal Characteristics". In: *Innovative Logistics Services and Sustainable Lifestyles*. Springer, 2019, pp. 87–104.

[232] Jane Kolodinsky and Jayson L. Lusk. "Mandatory labels can improve attitudes toward genetically engineered food". In: *Science Advances* 4.6 (2018). DOI: `10.1126/sciadv.aaq1413`.

[233] Rebecca G. Boswell et al. "Training in cognitive strategies reduces eating and improves food choice". In: *Proceedings of the National Academy of Sciences* 115.48 (2018), E11238–E11247. ISSN: 0027-8424. DOI: `10.1073/pnas.1717092115`.

[234] Guido Caniglia et al. "Experiments and evidence in sustainability science: A typology". In: *Journal of Cleaner Production* 169 (2017), pp. 39–47.

[235] Liran Christine Shan et al. "What makes smartphone games successful in food information communication?" In: *npj Science of Food* 4.1 (Jan. 30, 2020), p. 2. ISSN: 2396-8370. DOI: `10.1038/s41538-020-0062-8`.

[236] Pavel Senin. “Dynamic time warping algorithm review”. In: *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855.1-23 (2008), p. 40.

[237] Andrei Serjantov and George Danezis. “Towards an information theoretic metric for anonymity”. In: *International Workshop on Privacy Enhancing Technologies*. Springer. 2002, pp. 41–53.

[238] Gil Alterovitz et al. “Ontology engineering”. In: *Nature biotechnology* 28.2 (2010), pp. 128–130.

[239] *Drools: A Business Rules Management System*. `https://www.drools.org`. Accessed February 2020. 2020.

[240] *Social Impact Data Hack*. Available at https://sidh2017.ut.ee (last accessed: January 2020). 2017.

[241] Stephen C Johnson. “Hierarchical clustering schemes”. In: *Psychometrika* 32.3 (1967), pp. 241–254.

[242] Erèndira Rendòn et al. “Internal versus external cluster validation indexes”. In: *International Journal of computers and communications* 5.1 (2011), pp. 27–34.

[243] W Erwin Diewert. “Harmonized indexes of consumer prices: Their conceptual foundations”. In: *Swiss Journal of Economics and Statistics (SJES)* 138.IV (2002), pp. 547–637.

[244] Alex Polacco and Kayla Backes. “The amazon go concept: Implications, applications, and sustainability”. In: *Journal of Business and Management* 24.1 (2018), pp. 79–92.

[245] Peter Howson. “Tackling climate change with blockchain”. In: *Nature Climate Change* 9.9 (2019), pp. 644–645.

[246] Katharine Gammon. “Experimenting with blockchain: Can one technology boost both data integrity and patients’ pocketbooks?” In: *Nature Medicine* 24.4 (Apr. 1, 2018), pp. 378–381.

[247] Joshua Becker, Devon Brackbill, and Damon Centola. “Network dynamics of social influence in the wisdom of crowds”. In: *Proceedings of the national academy of sciences* 114.26 (2017), E5070–E5076.

[248] Winter Mason and Duncan J Watts. “Collaborative learning in networks”. In: *Proceedings of the National Academy of Sciences* 109.3 (2012), pp. 764–769.

[249] Richard P Mann and Dirk Helbing. “Optimal incentives for collective intelligence”. In: *Proceedings of the National Academy of Sciences* 114.20 (2017), pp. 5077–5082.

[250] Bernhard Bergmair et al. “Instantly Deployable Expert Knowledge - Networks of Knowledge Engines”. In: *arXiv* (2018). eprint: `1811.02964`.

[251] Aniket Kittur et al. “Scaling up analogical innovation with crowds and AI”. In: *Proceedings of the National Academy of Sciences* 116.6 (2019), pp. 1870–1877.

[252] Frank Figge, William Young, and Ralf Barkemeyer. "Sufficiency or efficiency to achieve lower resource consumption and emissions? The role of the rebound effect". In: *Journal of Cleaner Production* 69 (2014), pp. 216–224.

[253] Yi Zuo et al. "Personalized recommendation based on evolutionary multi-objective optimization [research frontier]". In: *IEEE Computational Intelligence Magazine* 10.1 (2015), pp. 52–62.

[254] TPL Nghiem and LR Carrasco. "Mobile applications to link sustainable consumption with impacts on the environment and biodiversity". In: *BioScience* 66.5 (2016), pp. 384–392.

[255] WJ Wouter Botzen and Jeroen CJM van den Bergh. "How sensitive is Nordhaus to Weitzman? Climate policy in DICE with an alternative damage function". In: *Economics Letters* 117.1 (2012), pp. 372–374.

[256] Ricardo Abejón et al. "Multi-Objective Optimization of Nutritional, Environmental and Economic Aspects of Diets Applied to the Spanish Context". In: *Foods* 9.11 (2020), p. 1677.

[257] Thibaut Lust and Jacques Teghem. "The multiobjective multidimensional knapsack problem: a survey and a new approach". In: *International Transactions in Operational Research* 19.4 (2012), pp. 495–520.

[258] Ariel Kulik and Hadas Shachnai. "There is no EPTAS for two-dimensional knapsack". In: *Information Processing Letters* 110.16 (2010), pp. 707–710.

[259] Hossein Zare and Masoud Hajarian. "Determination of regularization parameter via solving a multi-objective optimization problem". In: *Applied Numerical Mathematics* 156 (2020), pp. 542–554.

[260] Tobias Glasmachers, Tom Schaul, and Jürgen Schmidhuber. "A natural evolution strategy for multi-objective optimization". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2010, pp. 627–636.

[261] Kalyanmoy Deb and J Sundar. "Reference point based multi-objective optimization using evolutionary algorithms". In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. 2006, pp. 635–642.

[262] A Kelly, W Becker, and E Helsing. "Food balance sheets". In: *Food and health data: their use in nutrition policy-making. Copenhagen: WHO Regional Office for Europe* (1991), pp. 39–47.

[263] Kalyanmoy Deb et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE transactions on evolutionary computation* 6.2 (2002), pp. 182–197.

[264] Kalyanmoy Deb et al. "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". In: *International conference on parallel problem solving from nature*. Springer. 2000, pp. 849–858.

[265] Hisao Ishibuchi et al. "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations". In: *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE. 2009, pp. 1758–1763.

[266] Guillermo Fuertes et al. “Chaotic genetic algorithm and the effects of entropy in performance optimization”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.1 (2019), p. 013132.

[267] Hui Lu et al. “The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms”. In: *Mathematical Problems in Engineering* 2014 (2014).

[268] Xuefeng F Yan, Dezhao Z Chen, and Shangxu X Hu. “Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model”. In: *Computers & Chemical Engineering* 27.10 (2003), pp. 1393–1404.

[269] You Yong, Sheng Wanxing, and Wang Sunan. “Study of chaos genetic algorithms and its application in neural networks”. In: *2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM'02. Proceedings.* Vol. 1. IEEE. 2002, pp. 232–235.

[270] Ziwei Li and Sai Ravela. “Neural Networks as Geometric Chaotic Maps”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[271] Dan Hendrycks and Kevin Gimpel. “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv:1606.08415* (2016).

[272] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017, pp. 5998–6008.

[273] Victor Schmidt et al. *CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing*. `https://github.com/mlco2/codecarbon`. 2021. DOI: `10.5281/zenodo.4658424`.

[274] Kyuseop Kwak, Sri Devi Duvvuri, and Gary J Russell. “An analysis of assortment choice in grocery retailing”. In: *Journal of Retailing* 91.1 (2015), pp. 19–33.

[275] J. Bhatia and T. D. Breaux. “Towards an information type lexicon for privacy policies”. In: *2015 IEEE Eighth International Workshop on Requirements Engineering and Law (RELAW)*. 2015, pp. 19–24. DOI: `10.1109/RELAW.2015.7330207`.

[276] Aleecia M McDonald and Lorrie Faith Cranor. “The Cost of Reading Privacy Policies”. In: *A Journal of Law and Policy for the Information Society* 4.3 (2008), pp. 543–568.

[277] Evangelos Pournaras et al. *Self-regulatory information sharing in participatory social sensing*. Zurich, 2016. DOI: `10.1140/epjds/s13688-016-0074-4`. URL: `http://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-016-0074-4`.

[278] J. Soria-Comas et al. “Individual Differential Privacy: A Utility-Preserving Formulation of Differential Privacy Guarantees”. In: *IEEE Transactions on Information Forensics and Security* 12.6 (2017), pp. 1418–1429. ISSN: 1556-6013. DOI: `10.1109/TIFS.2017.2663337`.

[279] Yitao Duan and Yitao. "Privacy without noise". In: *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*. New York, New York, USA: ACM Press, 2009, p. 1517. ISBN: 9781605585123. DOI: 10.1145/1645953.1646160.

[280] Ronald L. Krutz and Russell Dean Vines. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Wiley Publishing, 2010.

[281] Klaus Kursawe, George Danezis, and Markulf Kohlweiss. "Privacy-Friendly Aggregation for the Smart-Grid". In: *Privacy Enhancing Technologies: 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings*. Ed. by Simone Fischer-Hubner and Nicholas Hopper. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 175–191. ISBN: 978-3-642-22263-4. DOI: 10.1007/978-3-642-22263-4_10.

[282] Alvin C. Burns and Ronald F. Bush. *Marketing research*. Pearson, 2014. ISBN: 9780133074673.

[283] Charu C. Aggarwal and Philip S. Yu. *Privacy-Preserving Data Mining*. Ed. by Charu C. Aggarwal and Philip S. Yu. Vol. 34. Advances in Database Systems. Boston, MA: Springer US, 2008. ISBN: 978-0-387-70991-8. DOI: 10.1007/978-0-387-70992-5.

[284] Tiancheng Li and Ninghui Li. "On the tradeoff between privacy and utility in data publishing". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*. New York, New York, USA: ACM Press, 2009, p. 517. ISBN: 9781605584959. DOI: 10.1145/1557019.1557079.

[285] Andreas Krause and Eric Horvitz. "A Utility-theoretic Approach to Privacy and Personalization". In: *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*. AAAI'08. Chicago, Illinois: AAAI Press, 2008, pp. 1181–1188. ISBN: 978-1-57735-368-3.

[286] Chao Li et al. "A Theory of Pricing Private Data". In: *ACM Transactions on Database Systems* 39.4 (2014), pp. 1–28. ISSN: 03625915. DOI: 10.1145/2691190.2691191.

[287] Craig Gentry. "Computing Arbitrary Functions of Encrypted Data". In: *Commununications of the ACM* 53.3 (2010), pp. 97–105. ISSN: 0001-0782. DOI: 10.1145/1666420.1666444.

[288] Cynthia Dwork. "Differential privacy". In: *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming* (2006), pp. 1–12. ISSN: 03029743. DOI: 10.1007/11787006_1. arXiv: arXiv:1011.1669v3.

[289] Ziteng Wang et al. *Efficient Algorithm for Privately Releasing Smooth Queries*. 2013. URL: https://papers.nips.cc/paper/5011-efficient-algorithm-for-privately-releasing-smooth-queries.

[290] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Foundations and Trends in Theoretical Computer Science* 9 (2014), pp. 211–407. ISSN: 1551-305X. DOI: 10.1561/0400000042.

[291] Ziteng Wang et al. “Differentially Private Data Releasing for Smooth Queries”. In: *Journal of Machine Learning Research* 17.51 (2016), pp. 1–42.

[292] P. Kairouz, S. Oh, and P. Viswanath. “The Composition Theorem for Differential Privacy”. In: *IEEE Transactions on Information Theory* 63.6 (2017), pp. 4037–4049. ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2685505.

[293] Reza Shokri and Vitaly Shmatikov. “Privacy-Preserving Deep Learning”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS ’15*. New York, New York, USA: ACM Press, 2015, pp. 1310–1321. ISBN: 9781450338325. DOI: 10.1145/2810103.2813687.

[294] Nhathai Phan et al. “Differential Privacy Preservation for Deep Auto-Encoders : An Application of Human Behavior Prediction”. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)* (2016), pp. 1309–1316.

[295] J. D. Gibson, B. Koo, and S. D. Gray. “Filtering of colored noise for speech enhancement and coding”. In: *IEEE Transactions on Signal Processing* 39.8 (1991), pp. 1732–1742. ISSN: 1053-587X. DOI: 10.1109/78.91144.

[296] Cynthia Dwork et al. “Exposed! A Survey of Attacks on Private Data”. In: *Annual Review of Statistics and Its Application* 4.1 (2017), pp. 61–84. DOI: 10.1146/annurev-statistics-060116-054123. eprint: https://doi.org/10.1146/annurev-statistics-060116-054123.

[297] P. Samarati and L. Sweeney. *Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression.* Tech. rep. Computer Science Laboratory, SRI Internation, 1998. URL: http://www.csl.sri.com/papers/sritr-98-04/.

[298] Jordi Nin, Javier Herranz, and Vicenç Torra. “On the disclosure risk of multivariate microaggregation”. In: *Data & Knowledge Engineering* 67.3 (2008), pp. 399–412. ISSN: 0169023X. DOI: 10.1016/j.datak.2008.06.014.

[299] Craig Gentry. “A fully homomorphic encryption scheme”. crypto.stanford.edu/craig. PhD thesis. Stanford University, 2009.

[300] Craig Gentry. “a Fully Homomorphic Encryption Scheme”. In: *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*. Vol. 19. STOC ’09 September. Bethesda, MD, USA: ACM, 2009, pp. 1–209. ISBN: 9781605585062. DOI: 10.1145/1536414.1536440.

[301] Craig Gentry and Shai Halevi. “Implementing Gentry’s Fully-Homomorphic Encryption Scheme”. In: *Advances in Cryptology – EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*. Ed. by Kenneth G. Paterson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 129–148. ISBN: 978-3-642-20465-4. DOI: 10.1007/978-3-642-20465-4_9.

[302] Yitao Duan. “Differential privacy for sum queries without external noise”. In: *ACM Conference on Information and Knowledge Management (CIKM)*. 2009.

[303] Andrew C. Yao. "Protocols for Secure Computations". In: *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*. SFCS '82. Washington, DC, USA: IEEE Computer Society, 1982, pp. 160–164. DOI: 10.1109/SFCS.1982.88.

[304] Stefano Bennati and Evangelos Pournaras. "Privacy-enhancing aggregation of Internet of Things data via sensors grouping". In: *Sustainable Cities and Society* 39 (2018), pp. 387–400. ISSN: 2210-6707. DOI: https://doi.org/10.1016/j.scs.2018.02.013.

[305] Wenliang Du and Mikhail J. Atallah. "Secure multi-party computation problems and their applications". In: *Proceedings of the 2001 workshop on New security paradigms - NSPW '01*. New York, New York, USA: ACM Press, 2001, p. 13. ISBN: 1581134576. DOI: 10.1145/508171.508174.

[306] Manoj Prabhakaran and Mike Rosulek. "Cryptographic Complexity of Multi-Party Computation Problems: Classifications and Separations". In: *Advances in Cryptology – CRYPTO 2008: 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*. Ed. by David Wagner. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 262–279. ISBN: 978-3-540-85174-5. DOI: 10.1007/978-3-540-85174-5_15.

[307] David Sánchez, Josep Domingo-Ferrer, and Sergio Martínez. "Improving the Utility of Differential Privacy via Univariate Microaggregation". In: *Lecture Notes in Computer Science*. Springer, Cham, 2014, pp. 130–142. DOI: 10.1007/978-3-319-11257-2_11.

[308] Ronald V. Book. "Comparing complexity classes". In: *Journal of Computer and System Sciences* 9.2 (1974), pp. 213–229. ISSN: 0022-0000. DOI: https://doi.org/10.1016/S0022-0000(74)80008-5.

[309] H. Nurmi. *Comparing Voting Systems*. Theory and Decision Library A: Springer Netherlands, 2012. ISBN: 9789400939851.

[310] Gleb Beliakov, Ana Pradera, and Tomasa Calvo. "Aggregation Functions: A Guide for Practitioners". In: *Studies in Fuzziness and Soft Computing*. 2007.

[311] E. Pournaras et al. "Engineering Democratization in Internet of Things Data Analytics". In: *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. 2017, pp. 994–1003. DOI: 10.1109/AINA.2017.15.

[312] Claude E Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.July 1928 (1948), pp. 379–423. ISSN: 07246811. DOI: 10.1145/584091.584093. arXiv: 9411012 [chao-dyn].

[313] Spyros Makridakis. "Accuracy measures: theoretical and practical concerns". In: *International Journal of Forecasting* 9.4 (1993), pp. 527–529. ISSN: 0169-2070. DOI: http://dx.doi.org/10.1016/0169-2070(93)90079-3.

[314] P. V. Rao, Eugene F. Schuster, and Ramon C. Littell. "Estimation of Shift and Center of Symmetry Based on Kolmogorov-Smirnov Statistics". In: *The Annals of Statistics* 3.4 (1975), pp. 862–873. ISSN: 00905364.

[315] P. M. Lerman. “Fitting Segmented Regression Models by Grid Search”. In: *Applied Statistics* 29.1 (1980), p. 77. ISSN: 00359254. DOI: 10.2307/2346413.

[316] Murray Spiegel. *Schaum’s Outline of Statistics*. 1992, p. 504. ISBN: 0071128204. DOI: 10.1036/0071485848.

[317] George CMA Ehrhardt, Matteo Marsili, and Fernando Vega-Redondo. “Phenomenological models of socioeconomic network dynamics”. In: *Physical Review E* 74.3 (2006), p. 036106.

[318] Silke Steingrube et al. “Self-organized adaptation of a simple neural circuit enables complex robot behaviour”. In: *Nature Physics* 6.3 (2010), pp. 224–230.

[319] LS Pontryagin et al. *Mathematical Theory of Optimal Processes [in Russian]*. Fizmatgiz Moscow, 1961.

[320] Murad Abu-Khalaf and Frank L Lewis. “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach”. In: *Automatica* 41.5 (2005), pp. 779–791.

[321] W Thomas Miller, Paul J Werbos, and Richard S Sutton. *Neural networks for control*. MIT press, 1995.

[322] Wanxin Jin et al. “Pontryagin Differentiable Programming: An End-to-End Learning and Control Framework”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[323] George Em Karniadakis et al. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440.

[324] Manuel A Roehrl et al. “Modeling system dynamics with physics-informed neural networks based on Lagrangian mechanics”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 9195–9200.

[325] Michael Lutter, Christian Ritter, and Jan Peters. “Deep lagrangian networks: Using physics as model prior for deep learning”. In: *arXiv preprint arXiv:1907.04490* (2019).

[326] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. “Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control”. In: *International Conference on Learning Representations*. 2020.

[327] Jie Sun and Adilson E Motter. “Controllability transition and nonlocality in network control”. In: *Physical Review Letters* 110.20 (2013), p. 208701.

[328] Atilim Gunes Baydin et al. “Automatic differentiation in machine learning: a survey”. In: *Journal of Machine Learning Research* 18 (2018).

[329] William L. Brogan. *Modern Control Theory (3rd Ed.)* USA: Prentice-Hall, Inc., 1991. ISBN: 0135897637.

[330] Seung-Yeal Ha, Hwa Kil Kim, and Sang Woo Ryoo. “Emergence of phase-locked states for the Kuramoto model in a large coupling regime”. In: *Communications in Mathematical Sciences* 14.4 (2016), pp. 1073–1091.

[331] Florian Dörfler, Michael Chertkov, and Francesco Bullo. "Synchronization in complex oscillator networks and smart grids". In: *Proceedings of the National Academy of Sciences* 110.6 (2013), pp. 2005–2010.

[332] Daoyi Dong. "Learning Control of Quantum Systems". In: *Encyclopedia of Systems and Control*. Ed. by John Baillieul and Tariq Samad. London: Springer London, 2020, pp. 1–7. ISBN: 978-1-4471-5102-9. DOI: 10.1007/978-1-4471-5102-9_100161-1.

[333] John R Dormand and Peter J Prince. "A family of embedded Runge-Kutta formulae". In: *Journal of Computational and Applied Mathematics* 6.1 (1980), pp. 19–26.

[334] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *IEEE International Conference on Computer Vision*. 2015, pp. 1026–1034.

[335] Keyulu Xu et al. "How Powerful are Graph Neural Networks?" In: *International Conference on Learning Representations*. OpenReview.net, 2019.

[336] Sylvie Thiébaux et al. "Decision-theoretic planning with non-Markovian rewards". In: *Journal of Artificial Intelligence Research* 25 (2006), pp. 17–74.

[337] Alberto Camacho et al. "Non-markovian rewards expressed in LTL: guiding search via reward shaping". In: *Annual Symposium on Combinatorial Search*. 2017.

[338] Eiji Mizutani and Stuart E Dreyfus. "Two stochastic dynamic programming problems by model-free actor-critic recurrent-network learning in non-Markovian settings". In: *IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*. Vol. 2. IEEE. 2004, pp. 1079–1084.

[339] Patrick Kidger, Ricky T. Q. Chen, and Terry Lyons. ""Hey, that's not an ODE": Faster ODE Adjoints with 12 Lines of Code". In: *arXiv preprint arXiv:2009.09457* (2020). arXiv: 2009.09457.

[340] National Research Council et al. "Recommended Dietary Allowances*". In: *Nutrition Reviews* 1.6 (1943), pp. 164–168. DOI: 10.1111/j.1753-4887.1943.tb08024.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1753-4887.1943.tb08024.x.

[341] *Compare sustainability of brands | buy sustainable | Rank a Brand*. https://rankabrand.org/. Accessed in April 2020. 2020.

[342] *GoodGuide*. https://www.goodguide.com/. Accessed in April 2020. 2020.

[343] *Shop Ethical! | Your ethical consumer guide*. https://www.ethical.org.au/3.4.2/. Accessed in April 2020. 2020.

[344] *Mobile Phone Ethical Comparison*. https://thegoodshoppingguide.com/ethical-mobile-phones. Accessed in April 2020. 2020.

[345] *The Green Stars Project | User-generated ratings for ethical consumerism*. https://greenstarsproject.org/. Accessed in April 2020. 2020.

[346] *Home - WikiRate*. https://wikirate.org/. Accessed in April 2020. 2020.

[347] *Behind the brands | Change the way the food companies that make your favorite brands do business.* Accessed in April 2020. 2020.

[348] Dirk Helbing. *Quantitative sociodynamics: stochastic methods and models of social interaction processes*. Springer Science & Business Media, 2010.

[349] Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. *The measurement of meaning*. 47. University of Illinois press, 1957.

[350] Parikshit Goswami and Tom O'Haire. "Developments in the use of green (biodegradable), recycled and biopolymer materials in technical nonwovens". In: *Advances in Technical Nonwovens*. Elsevier, 2016, pp. 97–114.

[351] G Villalba et al. "A proposal for quantifying the recyclability of materials". In: *Resources, Conservation and Recycling* 37.1 (2002), pp. 39–53.

[352] M. A. Rodriguez and M. J. Egenhofer. "Determining semantic similarity among entity classes from different ontologies". In: *IEEE Transactions on Knowledge and Data Engineering* 15.2 (2003), pp. 442–456. ISSN: 1041-4347. DOI: 10.1109/TKDE.2003.1185844.

[353] Norman Kretchmer. *Lactose and lactase*. 1972.

[354] G.J Miller and N.E Miller. "Plasma-high-density-lipoprotein Concentration And Development Of Ischaemic Heart-disease". In: *The Lancet* 305.7897 (1975). Originally published as Volume 1, Issue 7897, pp. 16–19. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(75)92376-4.

[355] Gian-Carlo Rota. "On the foundations of combinatorial theory I. Theory of Möbius functions". In: *Probability theory and related fields* 2.4 (1964), pp. 340–368.

[356] Dietmar Jannach & Markus Zanker & Alexander Felfernig & Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010. ISBN: 978-0-521-49336-9.

[357] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh. "Recommendation systems: Principles, methods and evaluation". In: *Egyptian Informatics Journal* 16.3 (2015), pp. 261–273. ISSN: 11108665. DOI: 10.1016/j.eij.2015.06.005.

[358] Hart Blanton et al. "When better-than-others compare upward: Choice of comparison and comparative evaluation as independent predictors of academic performance." In: *Journal of personality and social psychology* 76.3 (1999), p. 420.

[359] Wi-Suk Kwon, Basil Englis, and Manveer Mann. "Are third-party green–brown ratings believed?: The role of prior brand loyalty and environmental concern". In: *Journal of Business Research* 69.2 (2016), pp. 815–822.

[360] Béatrice Parguel, Florence Benoît-Moreau, and Fabrice Larceneux. "How sustainability ratings might deter 'greenwashing': A closer look at ethical corporate communication". In: *Journal of business ethics* 102.1 (2011), p. 15.

[361] George Papadimitriou, Athanasios Chatzidimitriou, and Dimitris Gizopoulos. "Adaptive Voltage/Frequency Scaling and Core Allocation for Balanced Energy and Performance on Multicore CPUs". In: *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2019, pp. 133–146.

[362] *Social Impact Data Hack 2017*. https://sidh2017.ut.ee/. [Online; April 2020].

[363] *nervousnet library*. https://github.com/prasad79/AssetDemoUsingAAR. [Online; April 2020].

[364] Evangelos Pournaras, Irina Moise, and Dirk Helbing. "Privacy-preserving ubiquitous social mining via modular and compositional virtual sensors". In: *In 2015IEEE 29th International Conference on Advanced Information Networking and Applications*. IEEE, 2015, pp. 332–338.

[365] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python ". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[366] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed in April 2020]. 2001. URL: `http://www.scipy.org/`.

[367] R. G. James, C. J. Ellison, and J. P. Crutchfield. "dit: a Python package for discrete information theory". In: *The Journal of Open Source Software* 3.25 (2018), p. 738. DOI: `https://doi.org/10.21105/joss.00738`.

[368] Andrei Novikov. *annoviko/pyclustering: pyclustering 0.8.2 release*. 2018. DOI: `10.5281/zenodo.1491324`.

[369] Plotly Technologies Inc. *Plotly, plotting library*. `https://plot.ly`. Montreal, QC, 2015. URL: `https://plot.ly`.

[370] Yadolah Dodge and Daniel Commenges. *The Oxford dictionary of statistical terms*. Oxford University Press on Demand, 2006.

[371] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax". In: *arXiv preprint arXiv:1611.01144* (2016).

[372] Penghang Yin et al. "Understanding straight-through estimator in training activation quantized neural nets". In: *arXiv preprint arXiv:1903.05662* (2019).

[373] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[374] J. Blank and K. Deb. "pymoo: Multi-Objective Optimization in Python". In: *IEEE Access* 8 (2020), pp. 89497–89509.

[375] Eckart Zitzler and Lothar Thiele. "Multiobjective optimization using evolutionary algorithms—a comparative case study". In: *International conference on parallel problem solving from nature*. Springer. 1998, pp. 292–301.

[376] Carlos M Fonseca, Luís Paquete, and Manuel López-Ibánez. "An improved dimension-sweep algorithm for the hypervolume indicator". In: *2006 IEEE international conference on evolutionary computation*. IEEE. 2006, pp. 1157–1163.

[377] Paul L Krapivsky and Sidney Redner. “Organization of growing random networks”. In: *Physical Review E* 63.6 (2001), p. 066123.