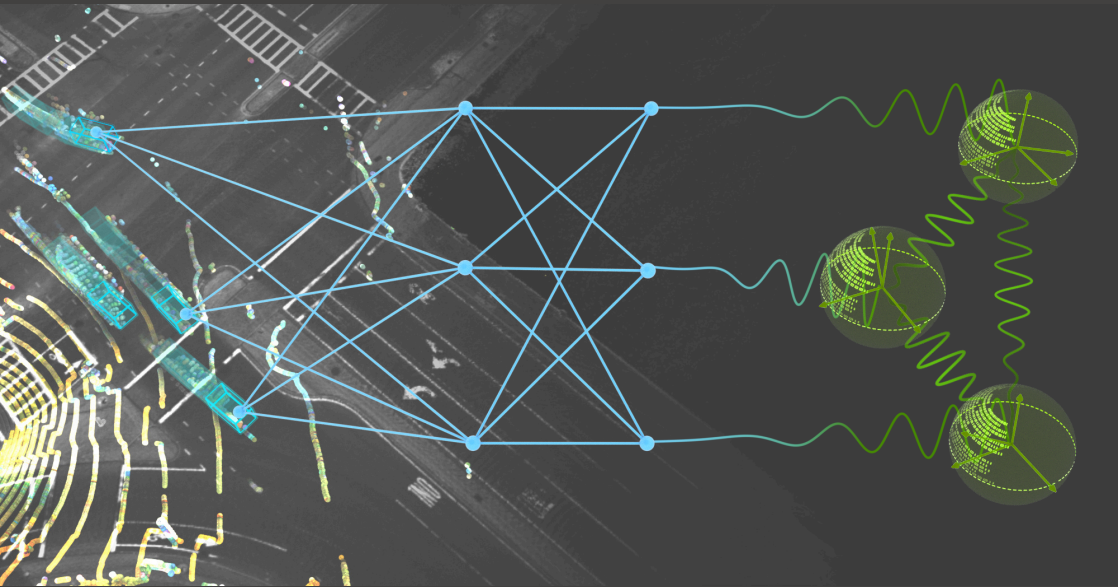


Vision for Autonomous Systems: From Tracking and Prediction to Quantum Computing

Ph.D. Thesis by Jan-Nico Zäch
Dissertation No. 29926 ETH Zürich



DISS. ETH NO. 29926

Vision for Autonomous Systems: From Tracking and Prediction to Quantum Computing

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES

(Dr. sc. ETH Zurich)

presented by

Jan-Nico Zäch

Master of Science in

Advanced Signal Processing and Communications Engineering

Friedrich-Alexander-Universität Erlangen-Nürnberg

born on 27/05/1993

accepted on the recommendation of

Prof. Dr. Luc Van Gool

Prof. Dr. Tat-Jun Chin

Dr. Colin Wilmott

Dr. Dengxin Dai

2024

ACKNOWLEDGEMENTS

First of all, I would like to thank Prof. Luc Van Gool for the opportunity to pursue my Ph.D. at the Computer Vision Lab and for always providing me with the freedom to investigate topics that were at the very heart of my interest. Especially his trust in my work on quantum computer vision has been important to me, as it allowed me to pursue a direction that had not yet been established in the community when I started my work. Furthermore, I am very thankful for his continuous support of the NomadZ RoboCup project which, throughout my Ph.D. developed from a lab-"duty" into a unique learning experience.

Similarly, I am very thankful for the strong support and close collaboration with Wim Abbeloos as my industry partner at Toyota Motor Europe. His early-on support for the topics that I have proposed has been exceptional and provided me with the very special opportunity to work on topics that were outside of the common scope of computer vision.

A major part of my Ph.D. journey has been strongly shaped by my advisors Martin Danelljan, Dengxin Dai, and Alex Liniger. From starting at ETH and finding the right topic for my Ph.D. all the way to bringing all of my work successfully together in this thesis. I am especially happy having worked together with Martin, who was quickly on board and deeply involved even with exotic research topics. This very quickly also became a great friendship and I will always be looking back at all the fun projects, conference trips and joint sailing holidays.

The most important part that helped me flourish during my Ph.D. has been the exceptional work environment at CVL that all of my colleagues have created. From working together late during deadlines to improvised conferences during a pandemic. I would like to especially mention my office neighbor Martin Hahner for helping me get started on the TRACE project, for our collaborative work, and for many fruitful discussions. Cadu Oliveira for helping me to prepare materials with large public outreach and getting another perspective on communicating research output. Goutam, Prune, and Mo for always creating a fun and inviting atmosphere and for being great company throughout the years at CVL.

Another important part of my Ph.D. has been the supervision of the NomadZ RoboCup project with the exceptionally motivated and talented students involved in it. Seeing such dedication to all aspects of the project,

from developing a fully autonomous software stack and participating in international competitions to communicating our results to many different audiences was truly inspiring. This would not have been possible without Giuliano Albanese who saved the project during the pandemic, Arka Mitra who played a major role in establishing research as a part of the project, Yan Wu and Lukas Molnar who considerably contributed to our research output, and Nicole Damblon, Zichong Li, Filippo Spinelli, Lucia Liu, and Koen Wolters for their work as part of the student team-lead.

Finally, all of this would not have been possible without the strong support of my family throughout my whole studies. By my mother Andrea who has always been there to support me, especially during times when things got stressful, my father Jörg who has provided me with the basis for my academic interests already early on, and my brother Luca who has been around for uplifting evenings whenever I visited back home. I will also always be deeply thankful for the encouragement I received from my grandparents who deeply believed in me accomplishing this big goal throughout my whole Ph.D. journey.

ABSTRACT

Autonomous systems strongly rely on computer vision to build a comprehensive model for understanding the environment they are embedded in. This task needs to be solved on multiple levels of abstraction, ranging from a high-level understanding of agent intentions to solving combinatorial problems for fundamental vision tasks. In this thesis, we focus on applications at three of these levels. On the most abstract level, we study the understanding of human intentions for autonomous driving and for a team of humanoid robots in structured environments. The foundation of this approach is multi-object tracking (MOT), which is subsequently investigated as one of the fundamental computer vision problems. Finally, on the lowest level of abstraction, we propose a quantum computing formulation of the matching problem our tracker is built on and further investigate the efficient use of an adiabatic quantum computer in computer vision.

In the *first part* of this thesis, the prediction of high-level actions of traffic participants in an autonomous driving scenario is studied. For this purpose, we develop a Hidden-Markov model representation that allows us to decode the sequence of actions from a vehicle’s trajectory and a semantic map present in large-scale driving datasets. For predicting future driving maneuvers, we propose a convolutional neural network that fuses map information and observed trajectories using a rendered representation.

We subsequently approach human action recognition from the perspective of an autonomous robot in a structured environment. To enable this, we collect a referee action dataset that contains multiple domains to cater to the requirements of the task. By using simulated images, the dataset can be adapted easily to new actions, while two kinds of realistic domains allow us to adapt to real images with a reduced annotation effort. We develop a computationally efficient network to detect the actions and deploy it on the humanoid NAO robot.

In the *second part*, we propose a learnable online 3D MOT approach that uses a predictive model for traffic participants together with deep learning-based object matching. To enable this, we define a graph structure that merges both representations and uses neural message passing to match pairs of detection at different timesteps as well as detections with tracks. We furthermore propose a two-stage training approach that models inference within an online system, while avoiding the expensive rollout of

online tracks. Overall, our method considerably improves track stability and performance.

After this, we further investigate improving long-term track stability on video sequences. This is done in the context of monitoring a fleet of robots from wide-angle cameras, where strong occlusions and identically looking robots pose a large challenge. We thus frame the task as a multi-platform sensor fusion approach, where tracklets from the external camera view are combined with measurements performed by the robots. The tracklets are combined into long-term tracks by solving a discrete quadratic problem that represents costs generated by different submodules. The cost weights are optimized using particle swarm optimization as a metaheuristic.

The *third part* of the thesis explores the application of quantum computing to challenging computer vision and machine learning tasks. We approach MOT with this paradigm by stating the matching and assignment problem as a task solvable on an adiabatic quantum computer (AQC). We further propose an iterative approach to represent and optimize the tracking constraints, for an improved solution probability. In simulation, we show that our approach is competitive with the state-of-the-art on commonly used MOT benchmarks. Using a D-Wave AQC, we demonstrate that small real-world problems can be solved on a quantum computer and provide an in-depth analysis of the properties of our approach using synthetic examples.

Finally, we approach the efficient use of an AQC for quantum computer vision and machine learning tasks. Starting from the perspective that many quantum computer vision applications are formulated as clustering tasks with additional constraints, we propose an approach that utilizes all measurements taken on an AQC to generate alternative high-quality clustering solutions. This uses the existing measurements to generate calibrated confidence scores for the solutions, with little additional compute cost. We validate our formulation with experiments in simulation and on a D-Wave AQC. Furthermore, we show that the set of solutions can be used to eliminate ambiguous points and that this approach also transfers to real data that does not strictly follow the assumptions of our derivation.

ZUSAMMENFASSUNG

Autonome Systeme sind in hohem Maße auf computerbasierten Sehens (Computer Vision) angewiesen, um ein umfassendes Modell zum Verständnis der Umgebung zu erstellen, in der sie eingebettet sind. Diese Aufgabe muss auf mehreren Ebenen der Abstraktion gelöst werden, von einem abstrakten Verständnis der Absichten aller Akteure bis hin zur Lösung kombinatorischer Probleme für grundlegende Aufgaben im computerbasierten Sehen. In dieser Arbeit konzentrieren wir uns auf Anwendungen auf drei dieser Ebenen. Auf der abstraktesten Ebene untersuchen wir das Verständnis menschlicher Absichten für autonomes Fahren und für ein Team von humanoiden Robotern in strukturierten Umgebungen. Die Grundlage dieses Ansatzes ist die Multi-Objekt-Verfolgung (Multi-Object Tracking (MOT)), die anschließend als eines der grundlegenden Probleme des computerbasierten Sehens untersucht wird. Schließlich schlagen wir auf der niedrigsten Abstraktionsebene eine Quantencomputing-Formulierung des Zuordnungsproblems vor, auf dem unser Tracker basiert, und untersuchen weiterhin den effizienten Einsatz eines adiabatischen Quantencomputers in computerbasiertem Sehen.

Im *ersten Teil* dieser Arbeit wird die Vorhersage von hochrangigen Aktionen von Verkehrsteilnehmern in einem autonomen Fahrscenario untersucht. Zu diesem Zweck entwickeln wir eine Darstellung mittels verdecktem Markowmodell (Hidden-Markov-Modell (HMM)), die es uns ermöglicht, die Sequenz von Aktionen aus der Trajektorie eines Fahrzeugs und einer semantischen Karte, die in groß angelegten Fahrdatensätzen vorhanden ist, zu erkennen. Zur Vorhersage zukünftiger Fahrmanöver schlagen wir ein faltungsbasiertes neuronales Netzwerk vor, das Karteninformationen und beobachtete Trajektorien unter Verwendung einer visuellen Darstellung fusioniert.

Anschließend betrachten wir die Erkennung menschlicher Aktionen aus der Perspektive eines autonomen Roboters in einer strukturierten Umgebung. Um dies zu ermöglichen, sammeln wir einen Schiedsrichter-Aktionsdatensatz, der mehrere Domänen enthält, um den Anforderungen der Aufgabe gerecht zu werden. Durch die Verwendung simulierter Bilder kann der Datensatz leicht an neue Aktionen angepasst werden, während zwei Arten realistischer Domänen es uns ermöglichen, uns mit reduziertem Annotationsaufwand an reale Bilder anzupassen. Wir entwickeln ein re-

chenleistungseffizientes Netzwerk zur Erkennung der Aktionen und setzen es auf dem humanoiden NAO-Roboter ein.

Im *zweiten Teil* schlagen wir einen lernbaren Online-3D-MOT-Ansatz vor, der ein Vorhersagemodell für Verkehrsteilnehmer zusammen mit einer auf Tiefenlernen basierenden Objektzuordnung verwendet. Dazu definieren wir eine Graphenstruktur, die beide Darstellungen zusammenführt und neuronalen Nachrichtenaustausch (Neural Message Passing) verwendet, um Paare von Objecten zu verschiedenen Zeitpunkten zu erkennen sowie mit bereits bestehenden Zielobjecten zu verbinden. Darüber hinaus schlagen wir einen zweistufigen Trainingsansatz vor, der die Inferenz innerhalb eines Online-Systems modelliert und dabei das kostspielige Ausrollen von verfolgten Objecten vermeidet. Insgesamt verbessert unsere Methode die Stabilität und Leistung der Verfolgung erheblich.

Danach untersuchen wir die Verbesserung der langfristigen Stabilität der Verfolgung in Videosequenzen. Dies geschieht im Kontext der Überwachung einer Flotte von Robotern aus Weitwinkelkameras, wo starke Verdeckungen und identisch aussehende Roboter eine große Herausforderung darstellen. Wir formulieren die Aufgabe daher als einen Multi-Plattform-Sensorfusionsansatz, bei dem verfolgte Objecte aus der externen Kameraperspektive mit Messungen kombiniert werden, die von den Robotern durchgeführt werden. Die kurzzeitig verfolgten Objecte werden durch die Lösung eines diskreten quadratischen Problems, das Kosten darstellt, die von verschiedenen Teilmodulen erzeugt werden, zu langfristig erkannten Objecten kombiniert. Die Kostengewichtung wird unter Verwendung der Partikelschwarmoptimierung als Metaheuristik optimiert.

Der *dritte Teil* der Arbeit erforscht die Anwendung des Quantenrechnens auf herausfordernde Aufgaben des computerbasierten Sehens und des maschinellen Lernens. Wir betrachten MOT mit diesem Paradigma, indem wir das Zuordnungs- und Zuweisungsproblem als eine Aufgabe formulieren, die auf einem adiabatischen Quantencomputer (AQC) lösbar ist. Wir schlagen weiterhin einen iterativen Ansatz vor, um die Randbedingungen darzustellen und zu optimieren, um eine verbesserte Lösungswahrscheinlichkeit zu erreichen. In der Simulation zeigen wir, dass unser Ansatz mit dem Stand der Technik auf gängigen Datensätzen konkurrenzfähig ist. Mit einem AQC der Firma D-Wave demonstrieren wir, dass kleine reale Probleme auf einem Quantencomputer gelöst werden können und bieten eine eingehende Analyse der Eigenschaften unseres Ansatzes unter Verwendung synthetischer Beispiele.

Abschließend nähern wir uns dem effizienten Einsatz eines AQC für Aufgaben des computerbasierten Sehens und des maschinellen Lernens. Ausgehend von der Perspektive, dass viele Anwendungen im Quantencomputerbasierten Sehen als Gruppierungsaufgaben mit zusätzlichen Bedingungen formuliert sind, schlagen wir einen Ansatz vor, der alle auf einem AQC vorgenommenen Messungen nutzt, um alternative hochwertige Gruppierungslösungen zu generieren. Dies nutzt die bereits vorhandenen Messungen, um kalibrierte Vertrauenswerte für die Lösungen zu generieren, mit geringen zusätzlichen Rechenkosten. Wir validieren unsere Formulierung mit Experimenten in Simulationen und auf einem D-Wave AQC. Darüber hinaus zeigen wir, dass die Menge der Lösungen verwendet werden kann, um mehrdeutige Punkte zu eliminieren, und dass dieser Ansatz auch auf reale Daten übertragen wird, die nicht strikt den Annahmen unserer Ableitung folgen.

PUBLICATIONS

The following publications are included in parts or in an extended version in this thesis:

- Jan-Nico Zaech, Dengxin Dai, Alexander Liniger, Luc Van Gool. Action Sequence Predictions of Vehicles in Urban Environments using Map and Social Context. IEEE/RSJ International Conference on Intelligent Robots and System (IROS), 2020.
- Arka Mitra*, Lukas Molnar*, Jan-Nico Zaech*, Yan Wu, Carlos Oliveira, Seonyeong Heo, Fisher Yu, Luc Van Gool. Multi-Domain Referee Dataset: Enabling Recognition of Referee Signals on Robotic Platform. IROS Workshop on Human Multi-Robot Interaction, 2023. * Joint first authors listed alphabetically.
- Jan-Nico Zaech, Alexander Liniger, Dengxin Dai, Martin Danelljan, Luc Van Gool. Learnable Online Graph Representations for 3D Multi-Object Tracking. IEEE Robotics and Automation Letters 7, 5103, 2022.
- Giuliano Albanese*, Arka Mitra*, Jan-Nico Zaech*, Yupeng Zhao*, Ajad Chhatkuli, Luc Van Gool. Optimizing Long-Term Player Tracking and Identification in NAO Robot Soccer by fusing Game-state and External Video. IEEE Winter Conference on Applications of Computer Vision (WACV), 2024. * Joint first authors listed alphabetically.
- Jan-Nico Zaech, Alexander Liniger, Martin Danelljan, Dengxin Dai, Luc Van Gool. Adiabatic Quantum Computing for Multi Object Tracking. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
- Jan-Nico Zaech, Martin Danelljan, Tolga Birdal, Luc Van Gool. Probabilistic Sampling of Balanced K-Means using Adiabatic Quantum Computing. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

Furthermore, the following publications were part of my Ph.D. research, but not included in this thesis.

- Peshal Agarwal, Danda Pani Paudel, Jan-Nico Zaech, Luc Van Gool. Unsupervised robust domain adaptation without source data. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022.
- Edoardo Mello Rella, Jan-Nico Zaech, Alexander Liniger, Luc Van Gool. Decoder fusion rnn: Context and interaction aware decoders for trajectory prediction. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.
- Gabriel Herl, Jochen Hiller, Mareike Thies, Jan-Nico Zaech, Mathias Unberath, Andreas Maier. Task-specific trajectory optimisation for twin-robotic x-ray tomography. IEEE Transactions on Computational Imaging, 2021.
- Jan-Nico Zaech, Dengxin Dai, Martin Hahner, Luc Van Gool. Texture underfitting for domain adaptation. IEEE Intelligent Transportation Systems Conference (ITSC), 2019.
- Martin Hahner, Dengxin Dai, Christos Sakaridis, Jan-Nico Zaech, Luc Van Gool. Semantic understanding of foggy scenes with purely synthetic data. IEEE Intelligent Transportation Systems Conference (ITSC), 2019.

CONTENTS

1	Introduction	1
1.1	Thesis Structure	4
Part I Action Recognition and Prediction		
1.I	Related Work	7
2	Action Sequence Predictions of Vehicles in Urban Environments using Map and Social Context	11
2.1	Introduction	11
2.2	Method	13
2.2.1	Network Architecture	13
2.2.2	Input and Output Representation	14
2.2.3	Probabilistic Action Predictions	15
2.3	Dataset	16
2.3.1	Argoverse	16
2.3.2	Map Information	16
2.3.3	Automatic Action Labeling	17
2.3.4	Test Data and Ordered Action Sequences	19
2.3.5	Dataset Statistics	20
2.4	Experiments	20
2.4.1	Training	20
2.4.2	Baseline Model	21
2.4.3	Evaluation Approach	22
2.5	Results	24
2.5.1	Direct Evaluation	24
2.5.2	N-Most Likely Ordered Sequences	24
2.5.3	Ablation Study	26
2.5.4	Dataset Scale study	27
2.6	Conclusion and Future Work	28
3	Multi-Domain Referee Dataset: Enabling Recognition of Referee Signals on Robotic Platforms	29
3.1	Introduction	29
3.2	Dataset Description	32
3.2.1	Referee Actions	33
3.2.2	Real Data - Test Setting	34
3.2.3	Synthetic Data	36

3.2.4	Real Data - Chroma Key	37
3.2.5	Data Split	39
3.3	Action Recognition	39
3.4	Experiments and Results	41
3.5	Conclusion	45
Part II Multi-Object Tracking		
II.I	Related Work	49
4	Learnable Online Graph Representations for 3D Multi-Object Tracking	53
4.1	Introduction	53
4.2	Method	55
4.2.1	Graph Representation of Online MOT	56
4.2.2	Neural Message Passing for Online Tracking	59
4.2.3	Training Approach	63
4.3	Experiments and Results	64
4.4	Conclusion	68
5	Optimizing Long-Term Robot Tracking with Multi-Platform Sensor Fusion	71
5.1	Introduction	71
5.2	Method	74
5.2.1	Data and Application	75
5.2.2	Camera Calibration and Pose Estimation	75
5.2.3	Multi Object Tracker	76
5.2.4	Jersey Color Detection	77
5.2.5	Robot States	77
5.2.6	Global optimization	78
5.2.7	Cost terms	79
5.2.8	Optimization of Cost Weighting	80
5.2.9	Reference Method: DeepSORT	81
5.3	Experimental Results	82
5.3.1	Ablation Study	83
5.3.2	Feature Importance	83
5.3.3	Explainability	84
5.4	Conclusion	87
Part III Quantum Computer Vision and Machine Learning		
III.I	Related Work	91
III.II	Basics of Quantum Computing	93
6	Adiabatic Quantum Computing for Multi-Object Tracking	97

6.1	Introduction	97
6.2	Quantum MOT	99
6.3	Traditional Solvers	105
6.3.1	Hessian Regularization	105
6.3.2	Post Processing	106
6.4	Experiments and Results	107
6.4.1	Lagrangian Multiplier	107
6.4.2	MOT ₁₅	111
6.5	Conclusion	113
7	Probabilistic Sampling of Balanced K-Means using Adiabatic Quantum Computing	115
7.1	Introduction	115
7.2	Theory	117
7.2.1	Energy-Based Models	117
7.2.2	Clustering	118
7.2.3	Clustering as QUBO	119
7.3	Probabilistic Quantum Clustering	120
7.3.1	Motivation	120
7.3.2	Data Model	121
7.3.3	Boltzmann Reparametrization	122
7.3.4	Maximum Pointsets	123
7.3.5	Inference Parameter Optimization	124
7.4	Experiments and Results	124
7.4.1	Solver Methods	124
7.4.2	Dataset and Metrics	125
7.4.3	Calibration Performance	126
7.4.4	Clustering Performance	127
7.4.5	Maximum Pointsets	128
7.4.6	IRIS Dataset	128
7.5	Conclusion	129
8	Conclusion	131
8.1	Summary of Contributions	131
8.2	Discussion, Limitations and Future Work	133
8.2.1	Action Sequence Predictions of Vehicles in Urban Environments	133
8.2.2	Recognition of Referee Signals on Robotic Platforms	134
8.2.3	Learnable Online Graph Representations for 3D MOT	136
8.2.4	Long-Term Robot Tracking with Multi-Platform Sensor Fusion	137

8.2.5	Adiabatic Quantum Computing for Multi-Object Tracking	137
8.2.6	Probabilistic Sampling of Balanced K-Means using AQC	138

Bibliography	141
--------------	-----

ACRONYMS

AP	Average Precision
AQC	Adiabatic Quantum Computing/Computer
CNN	Convolutional Neural Network
FPN	Feature Pyramid Network
HMM	Hidden Markov Model
IoU	Intersection over Union
k-NN	k-Nearest Neighbors
MLP	Multilayer Perceptron
MOT	Multi-Object Tracking
MPIR	Mean Player Identification Recall
NMS	Non-Maximum Suppression
PSO	Particle Swarm Optimization
QA	Quantum Annealing/Annealer
QUBO	Quadratic Unconstrained Binary Optimization
ReID	Person Re-Identification
SIM	Simulated Annealing
UMAP	Uniform Manifold Approximation and Projection

INTRODUCTION

Autonomous systems have sparked fascination in their users and designers for a long time. Ancient and early examples mostly include sophisticated automata used for entertainment and serving as curiosities. These machines, while not "autonomous" in the sense of today's systems, showcased the potential of using mechanical and later electrical systems to perform tasks without constant human intervention. They range from the Greek "Theater of Heron" and flying automata described by Mozi in China, to Al-Jazari's work in the "Book of Knowledge of Ingenious Mechanical Devices." This evolution peaked in the Renaissance period, notably with Leonardo da Vinci's inventions and drawings.

Later, the 20th century marked an important era for autonomous systems with the advent of computers and subsequently, the field of robotics. Scientists and inventors such as Alan Turing, who theorized about machine intelligence, and Norbert Wiener, the founder of cybernetics, laid the foundation leading to today's autonomous systems.

The later part of the century and the beginning of the 21st century brought the rise of machine learning, particularly deep learning, and increased computational power. These have revolutionized the capabilities of autonomous systems and lifted them to the level where they can be encountered in many daily situations. Today's autonomous vehicles, drones, and sophisticated robots build on top of vast amounts of data, use intricate algorithms, and have capabilities that span beyond hard-coded tasks. These systems can learn, adapt, and make decisions in real-time.

Even with their presence in daily situations, autonomous systems have not lost any of their enchantment and we see more and more autonomous systems interact directly with humans. However, to create a space that is shared by humans and robots on a large scale, a considerable number of challenges still need to be solved. Out of these, comprehensively understanding the environment is one of the most challenging and important ones as it is a key step towards operating safely. While it is very natural for a human to have situational awareness and to know what happens around oneself, it is hard to strictly and uniquely formalize this task and it needs to be approached on multiple levels of abstraction, which is the core theme of this thesis.

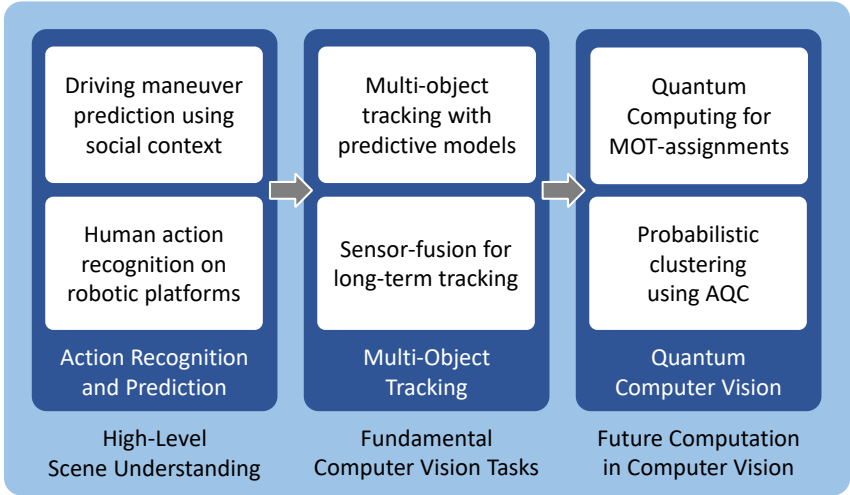


FIGURE 1.1: Overview of the thesis structure. The main theme is covering computer vision components in an autonomous system framework. Starting from high-level scene understanding the thesis advances towards more fundamental tasks of computer vision.

In the first part of the thesis, the highest level of understanding is approached, where the actions and intent of all agents in a scene must be detected and predicted. This information can be used within the autonomous framework to allow for long-term planning, to build a global state representing the scene as well as to provide interpretable information to any human working with the system. In the context of driving, an important responsibility for human drivers is to anticipate the actions of others, even if it often occurs unconsciously. This helps us to smoothly follow the traffic flow while keeping special attention and a sufficient distance from vehicles that are operated in an uncommon way. Furthermore, it allows one to adapt the driving style to a given scenario such as driving fast in areas where no risk stems from the observed intentions like in a well-predictable highway scenario and slowing down in uncertain scenarios like inner city traffic. In an autonomous vehicle, this information needs to be used similarly to plan a driving trajectory avoiding any dangerous situations proactively. The prediction problem often lacks a unique solution. Therefore, we model the task by predicting the probability of each potential action by a driver. While this high-level task is important for decision-making further down in a

whole autonomous driving framework, it also strongly builds on the prior steps of scene understanding and most importantly, a robust multi-object tracking pipeline, which forms the second part of this thesis.

Multi-object tracking aims to assign a unique identity to the same object over multiple frames in a time sequence. It is typically performed for a predefined set of object types relevant to autonomous driving scenarios, such as vehicles, pedestrians, and cyclists. While tracking is a fundamental building block for many autonomous systems, it is far from being solved in a robust way, especially in scenarios with strong occlusions or fast-moving objects. One approach towards solving this challenge is the integration of a model that represents any object that has been observed previously. Such a model can describe the dynamics and appearance of any object and in the simplest scenario be matched directly based on the distance between the predicted state and the state of a detected object. In this thesis, two major challenges in applying multi-object tracking to real-world scenarios are approached; the first work aims at integrating motion models and learned object matching into an online graph-based tracking framework using 3D LIDAR point clouds. This allows for a considerable improvement in track stability during occlusions and in cluttered scenarios. The second work investigates tracking of a fleet of robots, where long occlusions, identical appearance and clustering robots are the major challenges. We approach this by fusing the information perceived with each robot's own camera with an external camera view, which allows the tracker to follow the robots throughout a long video sequence. The core of this framework is a quadratic optimization problem for sensor fusion, which allows the merging of short tracklets into long tracks. However, similar to many discrete optimization problems in computer vision, this is hard to solve and constrained in its scale.

With quantum computing, a possible solution to this challenge is discussed in the last part of the thesis. Quantum algorithms promise a considerable speedup over their classical counterparts that are computationally prohibitive. Typical examples noteworthy in this context are Shor's prime factorization algorithm [205] and Grover's quantum search algorithm [81], which both improve the performance on these hard-to-solve problems. In computer vision, quantum computing has only recently been a topic of attention, and the works mostly focus on finding formulations that fit the architecture of a quantum computer. This often requires investigating the computational problem from scratch, as existing formulations are optimized to be efficiently solvable on classical hardware, which not necessarily is

the best formulation for a quantum-computing approach. Our work on quantum computing for multi-object tracking follows this approach by stating a frame-based formulation of the tracking assignment problem as a quadratic optimization problem. Furthermore, it investigates ways to modify the problem such that it can be solved with high probability and presents experimental results on a quantum annealer.

A second, still mostly open question in quantum computer vision is related to efficiently using a quantum computer, which most fundamental quantum computing research currently focuses on. On the other hand, quantum computer vision initially examines the application, aiming to enhance performance subsequently. In this thesis, the efficiency of quantum computing algorithms is approached for probabilistic k-means clustering. While most previous work discards all but the best solution measured on a quantum annealer, our work utilizes all measurements to provide a set of high-quality solutions together with calibrated confidence scores. This comes with little additional computational cost, as multiple measurements have to be performed in any scenario to assert a high probability of solving the task at hand.

1.1 THESIS STRUCTURE

The following parts of the thesis are structured according to Figure 1.1. The thesis follows the theme of vision for autonomous systems, where the main topics of **action prediction and understanding**, **multi-object tracking** and **quantum computing for computer vision** are covered. Each of the three topics is introduced with a section presenting related work, each followed by two chapters covering different topics that were investigated as part of this work. Furthermore, a chapter introducing the basics of quantum computing is included to make the thesis more accessible to readers with a computer vision background.

Part I

ACTION RECOGNITION AND PREDICTION

The first part of this thesis investigates high-level scene understanding, especially the task of understanding the intentions of human agents. Knowing the intention of humans allows autonomous agents to plan their actions accordingly, which ensures a safe operation in shared spaces. In Chapter 2, we investigate this task in the context of an autonomous vehicle. On a dataset of short trajectories that are recorded by an acquisition vehicle equipped with a large number of sensors [39], a method that uses semantic maps to extract high-level actions is presented. We show that a prediction model can be trained with this data and evaluate the limitations of current datasets. Chapter 3 studies human action recognition from the perspective of a fleet of humanoid robots, equipped with low-cost consumer level hardware. For this task, a multi-domain dataset is presented that aims at reducing data-collection and annotation costs. Furthermore, an action recognition method aiming at computational efficiency is developed and deployed on the humanoid NAO robot.

1.1 RELATED WORK

Behavior prediction formulated as trajectory forecasting for both humans and vehicles has been extensively studied. This extends to areas like predicting the high-level intentions of traffic agents and recognizing human actions in various scenarios.

The line of research closest to the method presented in Chapter 2 this thesis focuses on predicting high-level intention of traffic agents. It approaches driver action prediction of the ego vehicle where rich information about the state is available. Morris *et al.* [167] use rich sensor data including radar, lane marking detection and a head tracking camera to predict lane changes in a highway driving scenario. Jain *et al.* [105] extend this approach to a larger set of maneuvers and base their method on a video of the driver, maps, vehicle dynamics and an outside view in more diverse environments. In [88] surround video and map renderings are used to predict yaw and acceleration in an end-to-end framework. [201] forecast lane changes of other traffic agents in highway scenarios and analyze the challenge of heavily imbalanced data in this context. In more challenging urban environments, [118] classify driving actions at structured four-way intersections with an LSTM-based approach.

Early approaches to trajectories prediction, combine maneuver recognition with parametric motion models for each maneuver. Laugier *et al.* [130] use a Hidden Markov Model (HMM) with access to high-level information

such as distance to lane borders, signaling light status or proximity to an intersection to recognize behaviour of traffic agents. Trajectories are then sampled from a Gaussian process and used for evaluating collision risks. The same task is approached in [202] by detecting maneuvers with a Bayesian network. Houenou *et al.* [99] propose a heuristic maneuver detection module with full environment knowledge including each vehicle’s acceleration and yaw angle, together with an analytic description of trajectory sets. By using a variational Gaussian Mixture Model, Deo *et al.* [54, 55] implement probabilistic trajectory prediction for highway scenarios in the combined maneuver and trajectory prediction framework. Recently, [38] used a neural network-based approach to combine intention and trajectory prediction with dynamic HD maps and LIDAR information.

A second group of trajectory prediction algorithms directly approaches the task without intermediate state representations. Lee *et al.* [135] propose an end-to-end trainable recurrent neural network structure that includes scene context and samples multiple trajectories to capture the multi-modal nature of trajectory prediction. In [92], a wide range of output representations are evaluated in combination with a fully convolutional encoder structure. By defining a graph structure, [139] explicitly models the relation between multiple traffic agents and uses an LSTM-based encoder-decoder model to predict trajectories. Closely related to trajectory prediction, researchers at Waymo [14] learn a driving policy using rich maps and employ data augmentation to train robust models. By building a more strongly connected graph and learning relations subsequently, attention-based methods were able to further improve the utilization of agent relations [164, 189, 197]. Recently Scene Transformer [173] approaches the task of jointly predicting all traffic agents with a transformer architecture, which further strengthens the relationship modeling throughout the whole prediction pipeline.

In the context of pedestrian prediction and tracking, a central challenge is the modeling of interactions. Pellegrini *et al.* [179] show that social interactions and scene knowledge can boost tracking performance. [6, 82] predict trajectories based on past ego and social trajectory observations, while matching not only trajectories but also distributions. Sadeghian *et al.* [196] include world context using a top-down view and add attention to select the parts of the environment that are important. With a similar focus to our work, but in the context of pedestrian prediction, [210] explicitly investigates its multi-modal nature. Aiming at the interaction between a vehicle and pedestrian Zhang *et al.* [252] introduce a module into the network to specifically learn this behavior. Similar to predicting vehicle

trajectories, Transformers are used to model more complex social interaction among pedestrians [204, 228, 245, 251].

Parallel to this, human action recognition has been a cornerstone challenge in the computer vision community [24, 34, 60, 62, 107, 115, 206, 207, 216, 222, 227]. With the advent of deep learning, Simonyan and Zisserman [206] introduce a two-stream convolutional neural network for action recognition, laying the foundation of deep video action recognition. Subsequent works, such as the two-stream I3D [34], TSN [222], LRCNs [60] also make great progress on proposing innovative networks to capture the spatiotemporal features for video action recognition. Recently, the attention mechanism [78, 158, 169] is also introduced in the field of HAR. Furthermore, attention mechanisms were used as they are especially well suited for processing higher-level information such as skeleton data [182, 187].

For training these models, large human action recognition datasets are required. Besides a high annotation effort that is present for many tasks, they need to strongly account for privacy concerns when collecting humans performing a wide range of activities. UCF101 [207], HMDB51 [128], and Kinetics [115] are several widely-used large-scale video datasets for action training deep recognition networks, which cover a diverse set of human activities. However, collecting and annotating large-scale video datasets require tedious work, and therefore, synthetic datasets are also used to train visual models for many computer vision tasks [51, 74, 85, 156, 157, 218, 249].

Though models trained with synthetic datasets can show an acceptable performance when testing on real-world scenarios [74, 85, 156], the domain gap between the synthetic domain and the real domain remains to be an issue [180]. Two ways to approach this problem are domain adaptation [4, 101, 243, 247, 253] or the simulation and generation of more realistic data [84, 91, 218], which is the concept chosen in the method presented in Chapter 3.

ACTION SEQUENCE PREDICTIONS OF VEHICLES IN URBAN ENVIRONMENTS USING MAP AND SOCIAL CONTEXT

In this chapter, we study the problem of predicting the sequence of future actions for surrounding vehicles in real-world driving scenarios. To this aim, we make three main contributions. The first contribution is an automatic method to convert the trajectories recorded in real-world driving scenarios to action sequences with the help of HD maps. The method enables automatic dataset creation for this task from large-scale driving data. Our second contribution lies in applying the method to the well-known traffic agent tracking and prediction dataset Argoverse, resulting in 228,000 action sequences. Additionally, 2,245 action sequences were manually annotated for testing. The third contribution is to propose a novel action sequence prediction method by integrating past positions and velocities of the traffic agents, map information and social context into a single end-to-end trainable neural network. Our experiments prove the merit of the data creation method and the value of large automatically annotated datasets – prediction performance improves consistently with the size of the dataset and shows that our action prediction method outperforms comparing models.

2.1 INTRODUCTION

Autonomous driving is expected to fundamentally change our understanding of mobility and give us safer and more efficient traffic. One fundamental building block to achieve this is the ability to predict future actions of other road users. Only if one is able to accurately predict the potentially multi-modal future, collisions can be avoided. However, predicting future actions and trajectories of other road users requires a comprehensive understanding of traffic scenarios. This includes understanding the static and dynamic environment, as well as the traffic rules and the unwritten rules that govern how road user interact with each other.

The most common approach in this research direction is to directly predict the trajectories of other vehicles. While this is intuitive, allows for fully automatic data collection with today's test vehicles and yields a good

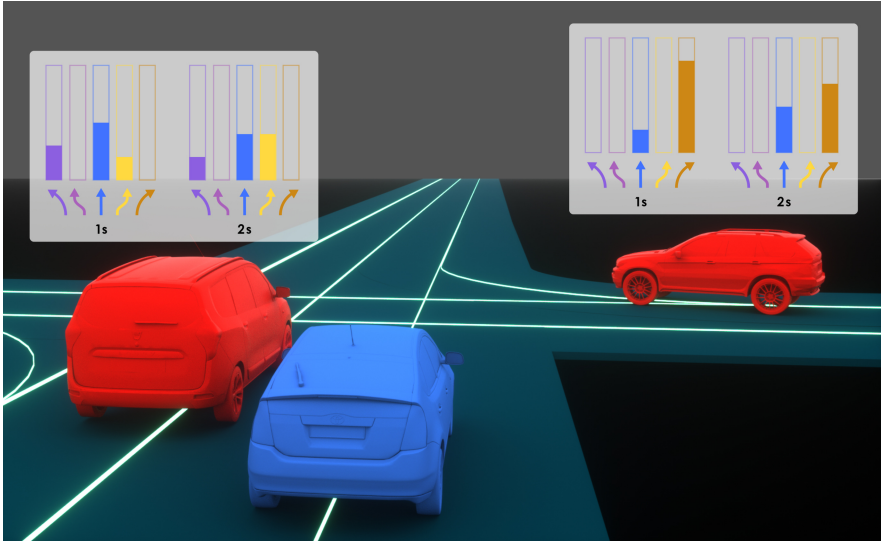


FIGURE 2.1: Concept of the proposed method: Based on trajectory and map information, high-level action sequences are predicted for surrounding vehicles. This representation is well interpretable and can potentially facilitate downstream planning tasks.

representation for decision-making and planning algorithms, it does not consider well that humans learn driving as a sequence of actions and also interpret driving scenarios that way. Furthermore, many traffic rules are defined as high-level representations using driving maneuvers. Thus an autonomous driving system requires emphasis on anticipating high-level actions of surrounding vehicles to better plan its own actions and to be more interpretable to humans.

To overcome the aforementioned challenges, we propose to state the problem of predicting traffic agents as the task of predicting action sequences and create a large-scale action prediction dataset based on real driving data. To circumvent the expenditure for manual annotations, we design an automatic method to convert trajectories of real-world traffic agents into action sequences. We apply the method to the large-scale dataset Argoverse [39] and compile a new action prediction dataset. The dataset contains 228,000 action sequences and features five distinct driving actions: *cruise* (\uparrow , c), *turn left* (\curvearrowright , tl), *turn right* (\curvearrowleft , tr), *lane change left* (\curvearrowright , ll) and *lane change right* (\curvearrowleft , lr) that describe normal vehicle operation. For testing, 2,245 trajectories from

the validation set have been manually annotated. The dataset allows us to train and compare models in a quantitative way in terms of their ability to predict a sequence of future actions.

We further propose an end-to-end trainable neural network that uses the past positions and velocities of the traffic agents, the map information and the social context to predict the future actions of the traffic agents. To fully leverage the power of convolutional neural networks (CNNs), we encode the information into a rendered image with multiple channels and use 2D and 3D CNN modules for prediction. As future actions are often uncertain and multimodal, our model outputs a probabilistic distribution of all plausible actions which can facilitate downstream planning. Fig. 2.1 showcases the concept of our action prediction approach.

2.2 METHOD

First, our method introduces an interpretable representation of traffic agents by forecasting high-level action sequences of a single traffic agent. Second, our method is completely learning-based and uses a sequence of maps, agents, as well as social information (other traffic agents) to predict the agent action sequence. This is done using a fully convolutional neural network with two-dimensional convolutions for computing spatial features and two three-dimensional layers for late and early fusion of temporal features. The network head is a fully connected layer that predicts the complete action sequence of the traffic agent at once with a single forward pass through the network. Finally, to make this method applicable for large-scale datasets, we introduce an automatic approach to generate action sequences from X-Y trajectory data and map information. Note that our method is *entity-centric* and only predicts the action sequence of one agent, however, we consider other traffic agents during prediction.

2.2.1 Network Architecture

We use a VGG-inspired architecture for action sequence prediction, which is shown in Fig. 2.2. Like [92], early and late fusion of temporal features is performed with three-dimensional convolutions. To avoid overfitting on the training data, we use a smaller model compared to some fully convolutional trajectory prediction approaches [92].

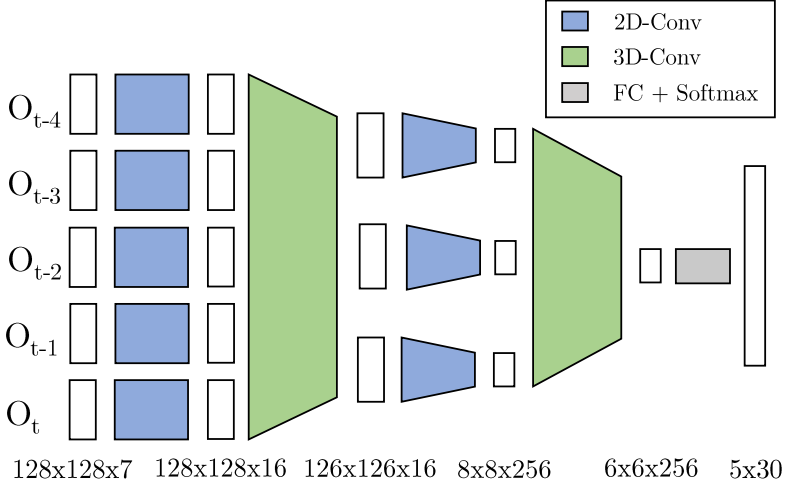


FIGURE 2.2: VGG inspired network architecture, with 2D convolutions to extract spatial features and 3D convolutions for early and late temporal fusion.

2.2.2 Input and Output Representation

The input to our neural network consists of three components: target agent, map, and social information. All the information is provided in a tensors with spatial dimension 128×128 and 7 channels at every timestep. We consider this combined information as our observation at time t , which we denote as \mathbf{O}_t . More precisely, \mathbf{O}_t is given as,

$$\mathbf{O}_t = (\mathbf{m}, \mathbf{1}_{target}, \mathbf{v}_{target}, \mathbf{1}_{other}, \mathbf{v}_{other})_t, \quad (2.1)$$

where, all components are rendered frames, spanning $50 \text{ m} \times 50 \text{ m}$. All frames are centered at the target agent's last observed position and rotated towards its driving direction. Each observation \mathbf{O}_t contains one layer for the rendered lane centerlines \mathbf{m} which stays fixed for all time steps. As the map only consists of centerline information, it can easily be captured with current industry grade perception systems and is present in most rich maps. Still, it naturally extends to more extensive information that might be available from manual annotations or more advanced data acquisition systems. Additionally, in \mathbf{O}_t the target agent is represented by three layers: $\mathbf{1}_{target,t}$, which is the indicator function representing the target agent's position by a one-hot encoded layer and two layers $\mathbf{v}_{target,t}$ representing

the current velocity in global coordinates. The other agents are represented the same way, but with all agents jointly rendered into the three available channels.

Altogether, the input to the network are the last five observations,

$$\mathbf{O} = (\mathbf{O}_{t-4}, \mathbf{O}_{t-3}, \mathbf{O}_{t-2}, \mathbf{O}_{t-1}, \mathbf{O}_{t_0}) , \quad (2.2)$$

where the observations are spread over the last two seconds, with $t \in (-2\text{ s}, -1.5\text{ s}, -1\text{ s}, -0.5\text{ s}, 0\text{ s})$.

To improve the performance and robustness of our method, data augmentation is used. More precisely, we rotate the network input randomly by θ , which is uniformly sampled from the range $-5^\circ \leq \theta \leq 5^\circ$. To perform the augmentation efficiently and without artifacts, all input data is stored in parametric form and rendered on the fly.

The output of the network is an independent probability distribution of the 5 action classes for 30 timesteps. With a sampling time of 100 ms, this results in a prediction horizon of 3 s. Note that the temporal relation and implicit dependence need to be learned by the model from the training data.

2.2.3 Probabilistic Action Predictions

In contrast to trajectory forecasting methods, our approach directly returns probabilistic predictions, without any requirement for sampling multiple forecasts [135] or defining spatially discretized grid-maps, as done in [92, 120]. Furthermore, this also allows for transparent performance measures since action classes can easily be interpreted by humans. The performance evaluation of trajectory prediction methods on the other hand normally uses averaged displacement errors. Even though this seems to be a transparent evaluation, due to imbalanced datasets, where following the current lane (our cruise action) is heavily over-represented, getting better displacement error does not necessarily imply that the method is better at forecasting the important corner cases. Note that the imbalance in the datasets can be massive, e.g. our dataset consists of 80% cruise states, however, there are other traffic environments where cruise trajectories can make up as much as 99% of the dataset [201]. This imbalance also explains that often simple constant velocity forecasting methods are competitive for short prediction horizons [92], even though they lack any understanding of the traffic scenario.

By predicting action sequences, our method does not directly solve the imbalance in the dataset. However, our evaluation method is more fine-grained and focuses on complex scenarios that require understanding the traffic scene.

2.3 DATASET

Many large-scale datasets for visual tasks in autonomous driving such as image segmentation, object detection or depth estimation have been released in recent years and fueled the development of corresponding learning-based methods [31, 39, 76, 116]. In contrast to this, modeling and prediction of human decision-making in driving scenarios is heavily underrepresented, which can be attributed to the lack of data. Public datasets, while being suitable for vision tasks, fall short when approaching the task of modeling human driving behavior in complex environments. Furthermore, most datasets directly intended for this purpose remain private [38, 92].

2.3.1 *Argoverse*

An exception is the Argoverse Trajectory Forecasting dataset [39], which contains approximately 325,000 automatically detected trajectories in an urban environment. Out of them, 245,000 cover 5s segments that can be used for our approach¹. The dataset further provides basic semantic map information, including lane centerlines and the drivable area. We augment this dataset by automatically annotating high-level actions such as lane changes and turns that are interpretable by humans and have the potential to boost low level tasks like trajectory forecasting [55]. Finally, we also automatically extract the velocity information of the agents, which further helps our prediction model.

2.3.2 *Map Information*

As shown and discussed in Section I.I, using HD-map information can be fundamental for traffic agent forecasting. However, we show that HD-maps can also be used to automatically annotate data, or in our case generate high-level action sequences from trajectories. This avoids time and labor-intensive manual labeling, and at approximately 245,000 trajectories in

¹ 80,000 trajectories are in the test set and only show 2s segments.

Argoverse, which corresponds to roughly 7.5M action annotations, this is the only viable approach.

On the one hand, relying on HD-maps for action sequence generation, is in our opinion not restrictive, since maps are regarded essential for the safe operation of autonomous vehicles. On the other hand, for large-scale datasets, annotating a semantic map with lane centerlines, which remains constant over time, scales favorably compared to annotating every time step of every agent recorded during driving. Furthermore, as current lane detection algorithms show impressive performance in a wide range of practical scenarios, automatic extraction of local map information could be feasible to further automate the labeling process.

For the task of action sequence annotation, the semantic HD-map available in [39] can be represented as a graph where each node corresponds to a short sequence of line segments and edges represent the relation between the line segments, which is visualized in Fig. 2.3. Edges can have the labels successor, predecessor, and neighbor lane-segment. Nodes contain the geometric properties describing the lane segments and semantic information if the segments describe a turn (either left or right) or a lane driving straight forward.

2.3.3 *Automatic Action Labeling*

Action extraction from trajectories is performed in a three-stage pipeline described in the following paragraphs.

TRAJECTORY SMOOTHING As the trajectories are generated from automatically detected objects, all samples are noisy and need to be filtered in the first stage. For this purpose, we use a bidirectional Kalman filter with a standard constant acceleration model, where the jerk is modeled as a noise input and we use the agent’s noisy position as the measurement. Note that this filter does not only help to smooth the agent’s position but also estimates the velocity of the agent.

LANE ASSIGNMENT In the second stage of the annotation pipeline, each sample from the trajectory is assigned to a node, which represents a lane segment in the graph map as shown in Fig. 2.3. Following the temporal order of samples, each trajectory induces a sequence of nodes that are visited. If the sequence of nodes follows a valid path through the graph, i.e. a path that only contains transitions between nodes that are connected

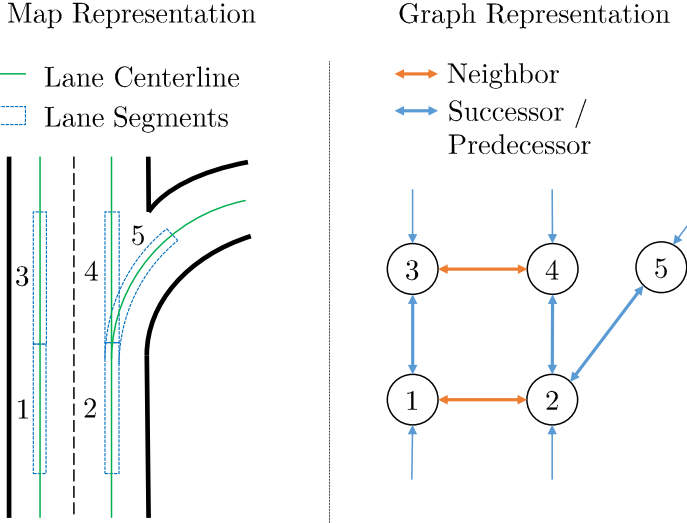


FIGURE 2.3: Map represented as rendered centerlines and as graph-map. Each centerline corresponds to a node in the graph-map which are connected by edges labeling their semantic relation.

by an edge, actions can be extracted from the graph. The property that only a valid path allows for the extraction of an action sequence form the base to design the node assignment algorithm. Using a purely geometric approach, where each sample is assigned to its nearest neighbor in an arbitrary measure, could lead to a large fraction of invalid trajectories. We thus propose to use a joint geometric and semantic lane assignment algorithm based on the Viterbi algorithm.

One can interpret a trajectory as the observation from an HMM defined by the graph map together with actions as latent variables. The graph-map induces a HMM, where the lane segment used by the agent at time step t corresponds to the hidden state X_t and its position $Y_t \in \mathbb{R}^2$ is the symbol emitted by the state. The emission probability

$$P(Y_t|X_t = x_t) \quad (2.3)$$

captures the driver's behavior of not perfectly following the centerline, uncertainties in the map and measurement noise. Using this viewpoint, the actions taken by the agent define the sequence of hidden states, or inversely: inferring the most likely sequence of hidden states from the

observed trajectory is equivalent to finding the underlying action sequence. This task can be solved with the Viterbi algorithm.

To reduce computational complexity, only lane segments that are closer than a threshold of 5 m to the observed trajectory are included as hidden states in the HMM. Prior knowledge about the driving behavior can be encoded in the transition matrix \mathbf{A} between lane segments, which we populate using the edges of the graph-map that describe the relation between lane segments. For the possible transitions to a successor, predecessor or neighbor lane, values of 1.0, 0.5 and 0.3 are assigned respectively. For transitions between non-connected lanes, a skew variable α is introduced. Setting $\alpha = 0$, yields state sequences that can always be annotated, given the map information is valid. This also includes trajectories that actually cannot be modeled by the 5 action classes used in our approach, such as U-turns. Thus, we set $\alpha = 0.001$ which allows for transitions between unconnected lane segments, preventing the annotation of an action sequence for cases that cannot be annotated with the given set of actions. It is important to note that for the transition matrix \mathbf{A} the sum of entries from a state does not necessarily sum up to 1 and thus, does not represent a traditional transition probability matrix. We use this representation to not penalize lane segments that have multiple lanes connected to them in the graph map, such that trajectories going through them do not come at a higher cost.

ACTION EXTRACTION With the most likely sequence of lane-segments determined by the Viterbi algorithm, actions can be extracted from the map information with a rule-based system. Lane changes are labeled for transitions between lane segments marked as neighbors² and turns can directly be labeled from the turn annotation in the lane segment.

All actions are handled as non-singular events, e.g. the lane change state is annotated for all time steps between leaving the previous lane and stabilizing on the new lane. To enable this, the smoothed trajectory together with the lane geometry is used to extract the start and end point of all maneuvers.

2.3.4 Test Data and Ordered Action Sequences

For testing, 2245 trajectories from the validation set have been manually annotated. However, the annotator did not generate temporal action sequences,

² Corner cases such as the direct change to the neighbor of a successor lane need to be modeled and handled adequately.

but what we call ordered action sequences. Whereas action sequences, as produced by our automatic labeling, contain an action for each time step, ordered action sequences only contain the order of actions. To make this clear, let us consider a simple left lane change example where the prediction horizon is six time steps. In our example, the action sequence is given by $\mathbf{a}_s = (c, c, ll, ll, c, c)$. This action sequence would correspond to the following order action sequence $\mathbf{a}_{os} = (c, ll, c)$, thus the exact temporal location of the lane change is lost, however, the gist of the maneuver is captured.

Annotating ordered action sequences is also significantly simpler and less error prone, compared to temporal action sequences. Thus, for our 2245 test trajectories, we have manually annotated ordered action sequences for the 3 s prediction horizon. Note that to avoid bias, the annotator had no access to the automatically generated labels.

2.3.5 Dataset Statistics

While the trajectories present in the Argoverse dataset are already filtered to show challenging behavior, the extracted action data is still imbalanced, with the majority of the samples representing the cruise class as shown in Table 2.1. While this mostly stems from the lower probability of encountering an active maneuver compared to just following a lane in cruise, it also reflects the fact that a turn or lane change is usually followed by the traffic agent stabilizing in the cruise state.

Whereas generating the dataset statistics for our automatically generated action sequences in the training and validation set is based on the number of occurrences, the statistics for the ordered action sequences use an adapted method. State proportions are estimated by counting states with the inverse number of total states annotated for the corresponding sequence, e.g. a sequence only annotated as cruise counts 100% towards cruise, while a sequence consisting of cruise and a following lane change counts with 50% towards both classes. In total, the state distribution for the train, validation, and test set is as shown in Table 2.1.

2.4 EXPERIMENTS

2.4.1 Training

Network parameters are optimized with the Adam optimizer and an initial learning rate of 10^{-4} . A step decay schedule with a stepsize of 10 epochs

Split	Total	Cruise	Turn		Change	
			left	right	left	right
train	191,841	84.5%	7.6%	4.0%	2.1%	1.9%
val	36,471	87.5%	4.0%	3.2%	2.9%	2.4%
manual	2,245	87.2%	4.1%	2.7%	3.5%	2.6%

TABLE 2.1: Label distribution of the annotated data.

and a factor of 0.5 is used for adapting the learning rate. Dropout of 0.5 is used to reduce overfitting. All variations of the network are trained for 50 epochs.

To compensate for the heavily imbalanced dataset, weighted random sampling is used for dataloading. Weights are assigned based on the presence of actions anywhere in the prediction horizon: if a turn is present, the sample gets weighted with a factor of 3, if a lane-change is present, the weighting factor is set to 10. This does not fully compensate for the imbalance but performed better than strictly weighting samples by their inverse probability.

Note that the positions and velocities of all traffic agents used in the input representation in Equation (2.1) are extracted from the noisy trajectory data with a bidirectional Kalman-filter, as described in Section 2.3.3. However, to ensure causality, the filter is only applied to the observed values.

2.4.2 Baseline Model

By proposing a new formulation of the traffic agent prediction task, no direct comparison to other methods is possible. However, trajectory forecasting methods are intended to predict a trajectory that represents the future actions of the agent. Therefore, it is possible to adapt a method originally designed for trajectory prediction to the new task. For our evaluation, a k-Nearest-Neighbor (k-NN) based method that was evaluated in [39]³ and outperformed all their tested deep learning models for multimodal prediction is adapted to our problem statement and used as a baseline-model for comparison. With the nearest neighbors in the training set, the automatically extracted action sequences can be used as a prediction for

³ arXiv:1911.02620 [v1] 6 Nov 2019

Method	Mean	Cruise	Turn		Change	
			left	right	left	right
random	20.0	87.5	4.0	3.2	2.9	2.4
k-NN (9) [39]	32.5	93.5	32.9	26.3	5.4	4.2
k-NN (50) [39]	36.9	95.0	39.2	34.7	8.2	7.1
k-NN (100) [39]	37.4	95.3	39.7	35.4	8.6	8.0
ours	61.4	97.8	67.9	68.4	33.5	39.4

TABLE 2.2: AP scores of the investigated methods on the automatically annotated validation set. Random sampling performance corresponds to the dataset proportion for each class. All scores in %.

the task at hand. Given the predictions from k-NN, the frequency of class labels at each time step results in a prediction that matches the probabilistic form of our proposed approach. We set $k \in \{9, 50, 100\}$ to compare the influence of sampling multiple trajectories and select the best model for further comparison.

2.4.3 Evaluation Approach

2.4.3.1 Direct Evaluation of Predictions

Interpreting every time step as an independent classification problem allows for the direct evaluation of the network’s performance. As the data is heavily imbalanced, classification accuracy on the whole dataset does not provide sufficient insight. Therefore, we measure performance by representing every action class as a binary classification problem, which allows the calculation of the average precision (AP) score. AP for each class and their unweighted average denoted as mean AP across all five action classes are used as a performance measure. This direct evaluation approach jointly measures the performance of predicting the right action classes together with the performance of predicting the right time step for a transition between two actions.

2.4.3.2 *N*-Most Likely Ordered Action Sequences

In addition to directly measuring the model’s precision on the temporal action predictions, we evaluate our method by extracting the N -most likely ordered sequences of actions, i.e. sequences that are only defined by the order of actions, not their exact length or transition times.

The extraction of the N -most likely ordered action sequences, is sensible as the number of different actions in the 3 s prediction horizon is small. For our test set only 10 samples, corresponding to 0.45% of the manual annotations, were labeled as a sequence of more than two actions. Therefore, for extracting ordered action sequences from the temporal action sequence predictions, we only consider sequences with at most two actions, which makes the approach tractable.

When extracting ordered action sequences, the model of independent actions used during training needs to be taken into account. While this assumption is a good model for the probability of an agent performing action a_t at time step t , it is not suitable to approximate the probability of observing a full action sequence $\mathbf{a}_s = (a_0, \dots, a_T)$. By just using the product of the predicted probabilities

$$p_{ind} = \prod_t \hat{p}_t^{a_t}, \quad (2.4)$$

as an estimate for the sequence probability, the high temporal correlation of actions is neglected. We thus model the probability of a sequence with the two actions (a_{b1}, a_{b2}) by

$$p_b(a_{b1}, a_{b2}, t_s) = \min(\hat{p}_{t_0}^{a_{b1}}, \dots, \hat{p}_{t_s}^{a_{b1}}) \min(\hat{p}_{t_s+1}^{a_{b2}}, \dots, \hat{p}_T^{a_{b2}}), \quad (2.5)$$

where the transition happens after time step t_s leading to the blocks $t_0 \leq t \leq t_s$ of a_{b1} and $t_s < t \leq T$ of a_{b2} . Given that within a block no transition may happen, the transition probabilities between identical actions are 1. Therefore, the total probability of the first block can be modeled as the lowest predicted probability for a_{b1} within $t_0 \leq t \leq t_s$

$$\min(\hat{p}_{t_0}^{a_{b1}}, \dots, \hat{p}_{t_s}^{a_{b1}}). \quad (2.6)$$

Note that the same holds for a_{b2} and the second block. Under the assumption that rare combinations, such as a left turn directly followed by a right turn, are assigned low probabilities by the network, the transition probabilities between the blocks are modeled as being equal for all action pairs (a_{b1}, a_{b2}) .

The problem of finding the most likely ordered sequence of two actions can then be defined as finding (a_{b1}, a_{b2}, t_s) that maximize the product of the two block probabilities

$$p_{b,opt} = \max_{t_s, a_{b1}, a_{b2}} p_b(a_{b1}, a_{b2}, t_s). \quad (2.7)$$

By repeatedly searching for the most likely sequence while suppressing already extracted action pairs (a_{b1}, a_{b2}) , the N -most likely ordered sequences can be extracted. We report total and per sequence top- N accuracy with $N \in \{1, 2, 3\}$ in Table 2.3 on manually annotated data. A sequence is rated as being detected if the ground truth ordered state sequence is one of the top- N predictions. Trajectories that are annotated with sequences of length > 2 are always treated as being predicted incorrectly when computing the total accuracy.

2.5 RESULTS

2.5.1 Direct Evaluation

Performance metrics for the direct evaluation of predictions for the k -NN baseline [39] and our proposed method are shown in Table 2.2. The results affirm that increasing the number of sampled nearest neighbors by one magnitude ($k=100$) compared to the usual setting can improve its performance. Across all methods, the AP for the cruise state is on a high level, followed by turn actions. Lane changes are harder to predict with AP values at a much lower level. Also, the largest relative improvement of the proposed method compared to the baseline implementation can be seen for the two lane change action classes.

This may be explained by the dataset statistics together with the shape of lane change trajectories. While lane changes as well as turns are underrepresented in the training data, turns have a much more distinct trajectory shape and thus can be detected more easily, even without modeling them explicitly.

2.5.2 N -Most Likely Ordered Sequences

Results of evaluating our proposed as well as the best-performing baseline method on the manually annotated ordered sequences are provided in Table 2.3. In contrast to the baseline model, where the cruise action class is
















Sequence	Ours			k-NN (100) [39]		
	Top1	Top2	Top3	Top1	Top2	Top3
total	82.5	89.8	93.0	81.2	86.4	88.6
c 	94.1	96.2	97.9	99.6	99.8	99.9
ll, c 	65.7	85.7	95.7	1.4	27.1	41.4
tl 	63.3	69.4	73.5	14.3	18.4	26.5
lr, c 	44.7	72.3	83.0	0.0	36.2	53.2
tl, c 	25.0	65.0	87.5	32.5	95.0	97.5
tr, c 	42.5	75.0	80.0	17.5	70.0	87.5
c, tl 	12.1	69.7	75.8	0.0	42.4	69.7
c, tr 	24.2	72.7	78.8	0.0	36.4	48.5
c, ll 	4.3	52.2	73.9	0.0	17.4	26.1
ll 	23.8	38.1	42.9	0.0	0.0	0.0
lr 	11.1	22.2	22.2	0.0	0.0	0.0
tr 	35.3	47.1	58.8	0.0	0.0	5.9
c, lr 	6.3	75.0	75.0	0.0	6.3	6.3
tr, ll 	14.3	21.4	42.9	0.0	0.0	0.0
tl, lr 	40.0	50.0	60.0	0.0	0.0	10.0

TABLE 2.3: Evaluation of the top- N ordered sequence predictions. The shown sequences are ordered by descending frequency and at least have 10 samples. All scores are accuracies in %.

overrepresented, our proposed method predicts a more diverse set of action sequences. While a relatively high top1-accuracy is observed for sequences that require detecting a maneuver, e.g. (tl), (ll, c), (lr, c), sequences that involve the prediction of maneuvers such as (c, ll) or (c, lr) show a high improvement in the top2-accuracy. The observation that some action classes heavily improve in top2-accuracy is in line with the commonly accepted assumption that agent prediction needs to be multimodal to account for the uncertainty of the future.

Fig. 2.4 and 2.5 show the confusion matrix for the top-1 and top-2 sequence prediction, with sequences ordered according to their frequency in the dataset. For top-1 prediction, the classification errors mostly stem from assigning sequences with an incorrect second action, e.g. (tl) instead of (tl,

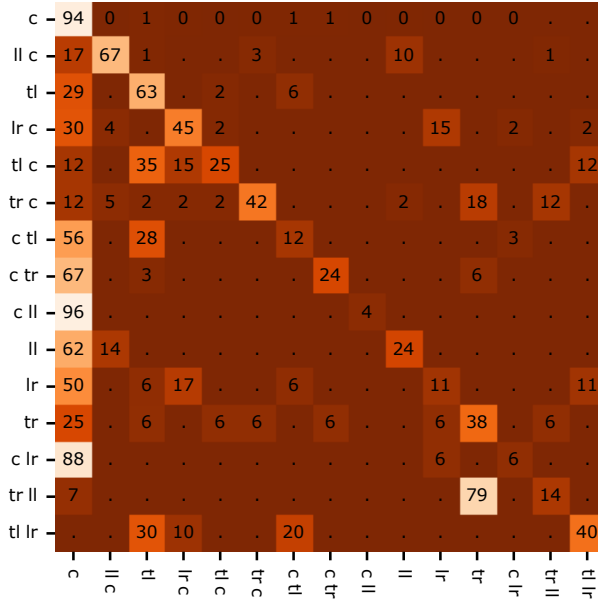


FIGURE 2.4: Confusion matrix for top-1 ordered sequence predictions with the x-axis showing predicted classes and y-axis showing ground truth classes. All numbers in %, normalized that rows sum to 100.

c), which confirms that prediction is a much harder task than classification. Still, the top-2 predictions allow for correcting for many of these errors, resulting in substantially higher values on the diagonal.

2.5.3 Ablation Study

An ablation study is conducted to evaluate the impact of the separate modules. Ablated methods are compared using the mean AP on the validation dataset, with results shown in Table 2.4. To investigate the influence of input representation the input tensor is grouped into the following modules: The target agent information (Target) includes positions and velocities of the target agent ($\mathbf{1}_{target}, \mathbf{v}_{target}$). Social context (Social) comprises the positions and velocities of the other agents ($\mathbf{1}_{other}, \mathbf{v}_{other}$). Finally, map information (Map) contains the layer representing the lane centerlines \mathbf{m} .

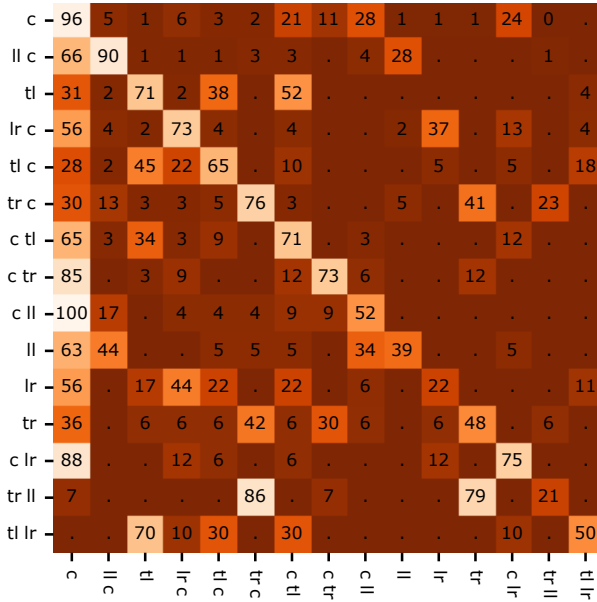


FIGURE 2.5: Confusion matrix for top-2 ordered sequence predictions with the x-axis showing predicted classes and y-axis showing ground truth classes. All numbers in %, normalized that rows sum to 200.

Besides ablating the input representation, we show that traffic agent action prediction can profit from the presented data augmentation approach of rotating the complete input representation by a small random angle.

The results confirm the usefulness of providing all three sources of information jointly, with the map having the biggest impact. Interestingly, adding social information to a representation that does not contain a map, has a substantially higher impact than adding it to a representation that does contain a map. This indicates that there is redundant information available in the map and social context the network is able to use.

2.5.4 Dataset Scale study

The second ablation study investigates the relevance of the dataset size for the action prediction task while using the full model. Thus, the network is trained with three subsets of our data that reflect 50%, 10% and 5% of the full set of action sequences. The dataset reduction is implemented by

Input representation			Augmentation	mean AP
Target	Social	Map		
✓			✓	0.375
✓	✓		✓	0.490
✓		✓	✓	0.602
✓	✓	✓		0.590
✓	✓	✓	✓	0.614

TABLE 2.4: Ablation study on the action prediction network.

Dataset proportion	mean AP
100%	0.614
50%	0.571
10%	0.522
5%	0.493

TABLE 2.5: Ablation study on the action prediction dataset.

skipping the corresponding number of samples, to avoid changing the dataset statistics. Epoch length and sample weights are kept constant, such that the training parameters are stable. The results of the ablation study, shown in Table 2.5, show a large difference in performance between the different scales, indicating that the amount of data has a major impact on the prediction performance. Furthermore, the considerable improvement from 50% of the data to the full dataset allows for the conclusion that the performance did not saturate at the scale present in the Argoverse dataset and larger amounts of data could benefit the community.

2.6 CONCLUSION AND FUTURE WORK

In this chapter, we investigated the task of predicting high-level actions of vehicles in urban environments. To make this task viable, we proposed an algorithm to automatically extract action sequences using HD-maps, from the public Argoverse [39] dataset. Furthermore, we proposed an action prediction network, that predicts the future action sequence considering agent, map, and social information. The network is completely based in rendered images and can be trained in an end-to-end fashion using the automatically generated large-scale action sequence dataset. We showed that our action prediction model, together with our dataset, can outperform existing methods that are adapted from trajectory prediction.

MULTI-DOMAIN REFEREE DATASET: ENABLING RECOGNITION OF REFEREE SIGNALS ON ROBOTIC PLATFORMS

Recognizing referee signals is a key aspect of playing soccer games with human players as well as in RoboCup, where Robots compete with each other. In these games, the current development prioritizes making the robots fully autonomous, where a key aspect is to interact with humans by understanding and interpreting signals provided by a referee. To cater to this, we present the Multi-Domain Referee Dataset in this chapter. The dataset aims to spur the development of high-efficiency action recognition methods in RoboCup as well as provide a basis to study the transfer between simulated and real domains in a strongly structured environment. To this end, we provide 3,108 action sequences with more than 183,000 images in total, spanning four domains: one fully synthetic, two hybrid ones that combine real images with synthetic augmentations, and one real domain for testing. To study the properties of the multi-domain dataset, we develop a recognition model capable of real-time inference inside a robotic framework on the Intel-Atom-based NAO robot. Our experiments show that combining real and synthetic data considerably improves performance and that new signals and settings can be learned efficiently by only updating the synthetic data, which can reduce the acquisition effort incurred by future rule changes in RoboCup.

3.1 INTRODUCTION

The RoboCup competition serves as a testing ground for autonomous systems, requiring teams to investigate all scientific aspects related to safely operating a robot fleet. This ranges from building robots over developing robust vision algorithms to learning global game strategies to engage in soccer matches autonomously. This all serves the overall objective of playing against the human world cup winners in the middle of the 21st century following official FIFA rules [125]. One critical component required for achieving this goal is the capability to understand the same signals a human player can use, with a special focus on referee actions. These can

signal the game's state as well as events such as fouls, offside situations, or penalty kicks and have a direct impact on the game's dynamics and outcomes. Thus, an accurate and efficient referee action recognition system is essential for the success of RoboCup teams.

Recently the community around the Standard Platform League has identified this research aspect to be crucial for moving the state-of-the-art forward and held multiple research challenges focused on this aspect. However, the achieved performance is behind expectations from current general action recognition methods and varies strongly between teams¹. This can be attributed to the distinct challenges faced in the RoboCup environment as well as the lack of a common dataset that can be used for training recognition models.

With the dataset and method described in this section, we provide a basis for approaching this goal and investigate the challenges of referee action recognition within the context of RoboCup. We explore the unique constraints and opportunities presented by this domain and present a comprehensive dataset to facilitate the development and evaluation of referee action recognition in future RoboCup tournaments. Our dataset is designed to encompass the complexities of real-world RoboCup matches while providing teams with the necessary resources to overcome the inherent challenges.

Compared to the generally used definition of human action recognition in literature [24, 62, 115, 206, 207, 216, 227], referee action recognition in RoboCup comes with its own distinct set of challenges and limitations. These are set by the environment as well as by the robots used during the matches. Their main objective is to provide an affordable humanoid robotic platform, which entails the following limitations:

- **Low Cost:** The robots are equipped with low-quality cameras that have to capture referee actions in suboptimal conditions like varying lighting and occlusions.
- **Low Latency Constraint:** Real-time performance is crucial on a robotic platform, imposing strong latency requirements on the recognition algorithm.
- **Low Compute Capability:** The compute resources available on the robots are strongly limited, necessitating lightweight, yet accurate recognition models.

¹ <https://spl.robocup.org/>

- **Time and Cost Effective Data Acquisition:** Teams have restricted resources that need to be distributed over all parts of the development. This limits the ability to acquire large-scale, diverse datasets for training and evaluation for all tasks by each team separately.

Nevertheless, at the same time, the setting used in RoboCup offers distinct properties that can be exploited when designing action recognition algorithms:

- **Multi-Platform Sensor Fusion:** Multiple robots are deployed on the field. With all of them being able to observe the referee, sensor fusion on multiple platforms can be employed to improve recognition accuracy.
- **Consistent Game Layout:** The game layout remains consistent across games. The robots are identical for all teams and the field is only scaled moderately between lab settings and real world-cup games.
- **Existing Robotic Framework:** A robotic framework with all modules to perceive and interact with the environment is available. This provides the foundation for easily deploy referee action recognition on the robot.

To address these challenges and leverage the opportunities, we present a dataset that not only models the full range of referee actions used in the tournament but also caters to the limitations and strengths of the RoboCup environment. The main contributions presented in this Section are as follows:

- A multi-domain dataset for referee action recognition in RoboCup games, encompassing synthetic data, real data captured against a chroma key background with synthetic augmentations, and real data collected in multiple environments.
- An action recognition pipeline that utilizes the dataset for training and evaluating the algorithm, demonstrating its potential as a benchmark for this task.
- Experimental results showcasing the effectiveness of the multi-domain dataset in improving recognition performance beyond using a single domain only.

Including data from multiple domains (real, real-chromakey, synthetic) allows us to quantify the transfer-capability between domains on this strongly

structured task as well as allows us to scale the amount of available data considerably. This is especially important in the context of the dynamically evolving RoboCup tournament. With changing locations, referee signals and referee outfits, synthetic data provides a way to generate new data in a cost- and time-efficient way.

Furthermore, strong control over the synthetic data generation pipeline ensures that the annotations are correct; unlike real-life conditions where ambiguity in the data might occur due to noise in the human annotations. Finally, real-world datasets also raise privacy concerns; where the individual participants need to agree to their capture for research purposes. Synthetic data circumvents these ethical dilemmas, making it faster and easier to obtain and use.

3.2 DATASET DESCRIPTION

The dataset aims at enabling referee gesture detection on mobile robots. To this end, we provide a dataset that contains rendered **synthetic** videos, two sets of videos with a **chroma key** background and different acquisition protocols, and a set of **real** videos for benchmarking in a realistic setting.

The challenge we provide data for has first been presented at RoboCup 2022, where one robot was placed at the center circle of a soccer field, while the referee was standing in front of the robot. The robot had to detect a whistle that initiated the action detection. In a later version of this challenge (2023), the referee stands at a predefined position to perform a gesture, while the robots can be located anywhere on the field. The robots' positions represent a random game state, as the action is performed during a competitive match, again indicated by a whistle. The challenge is far from being solved and each year new additions are made to close the gap to referee actions in actual soccer games.

Our robotic pipeline deployed in RoboCup captures and processes videos at 15 frames per second, and annotations are done with single-frame precision. In addition, synchronization of multiple robots needs to be performed manually due to the possible frame drops within the real robotic framework. All these challenges make the task of gathering and annotating a large-scale dataset for referee action recognition very time-consuming.

As the robots used in the Robocup tournaments are well-documented and the physics that they follow can be clearly modeled, a similar environment can be configured in a simulated environment. This approach not only offers controlled data generation capabilities for researchers but

also enables us to generate large-scale datasets much more efficiently. In addition, given that the actions employed in the tournament may evolve from year to year, simulations provide a flexible means of adapting to new gestures introduced annually, with minimal adjustments required in the data generation pipeline.

Nevertheless, synthetic simulators often struggle to replicate the intricacies presented in real-life scenarios, leading to the presence of a typical domain gap between synthetic and real data. Green screens provide one way to bridge the domain gap between the simulated and real-life data. By allowing the background to be changed, green screen data enables strong data augmentation by recording a single real session in a well-defined environment. We leveraged the collected green-screen data and generated synthetic data for action recognition model training, and to test its performance in real-life scenarios, we also collected video sequences in real environments. Details about the gestures and the process for the data-collection of the data are described in detail in the following.

3.2.1 *Referee Actions*

The flow of soccer games is complex and controlled by the referee. With many events that can stop or alter the gameplay, understanding these events from the robot's perspective is crucial to playing autonomously without human interaction. When critical events occur in RoboCup, the referee performs different static or dynamic gestures after indicating them by the whistle. Often, these gestures also show which team the event is attributed to, which results in a range of gestures that can be performed symmetrically in two directions, where the most extended hand indicates the direction.

Figure 3.1 shows the action defined for the RoboCup 2023 and the corresponding event they refer to. As all but one action exists in two directions, we only show a single version of each action. Most actions are static and depicted in Figure 3.1a.

The last two dynamic actions consist of a dynamic motion followed by a static pose to denote which team needs to be considered. These are demonstrated in Figure 3.1b. Furthermore, the dynamic action classes 12 and 13 can be identified by considering the dynamic part and the static pose. The dynamic part is identical and the static action is similar to the action performed during kick-in. However, depending on the action-recognition approach these can also be detected separately with the provided dataset.

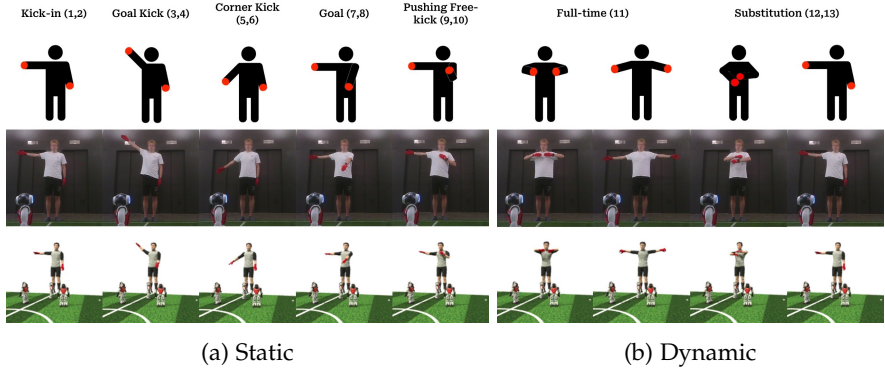


FIGURE 3.1: Referee Actions in the dataset. The first row represents the ideal pose. The second row is the same action performed by a real human. Synthetic pose is shown in the third row. Most gestures have a pair of actions where the arm reaches in opposite directions and only one direction is displayed here.

3.2.2 Real Data - Test Setting

We collect data from the robot cameras at 6 different locations that cover a variety of backgrounds and lighting conditions representative of environments present during RoboCup. To record a single session, the robots are randomly placed on the field with all robots facing the referee. The referee performs all 12 actions sequentially, separated by short breaks, while the robots record the video at 15 fps. We furthermore, collect data with the referee performing random motions, which we label as undefined motion (class 0). This data is solely used in the test set, to evaluate how our models trained on other data domains generalize to real-world settings.

DATA ANNOTATION Data is collected by multiple robots in parallel, and by synchronizing these robots, the data annotation can be accelerated. After initializing each robot, its internal clock precisely measures the starting position of the recorded video sequences. However, in certain cases, the internal clocks of some robots may deviate over time. To account for this, the starting time of each action in the action sequences is manually annotated. This annotation serves to correct for any time drift that may occur when running multiple sessions sequentially.

Action sequences captured by the robot with the clearest view are annotated manually with single-frame accuracy. Subsequently, a 60-frame

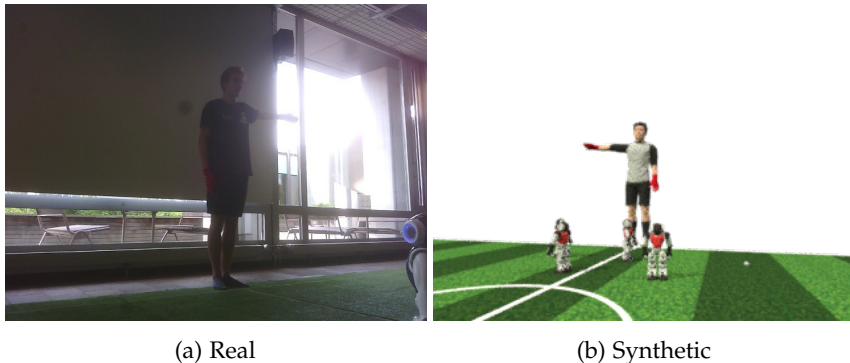


FIGURE 3.2: Dataset examples.

(4-second) interval is extracted for each action from the action sequences, starting from the annotated initial action pose. This duration is long enough to cover a complete action sequence. By manually synchronizing all the robots, the annotation process for data collected by other robots can be easily conducted as well.

The data collected with the real robotic platform introduces more challenges in achieving accurate action recognition, some of which are even difficult for humans to recognize. Several factors contribute to these challenges, including wear and tear in the cameras of certain robots and some challenging natural lighting conditions. These conditions encompass issues such as a green tint, underexposure, referees partially exiting the frame, and high-exposure backgrounds (one such example is shown in Figure 3.2a). In light of these challenges, recorded sequences are further manually rated and divided into easy and hard categories, indicating the challenge of accurately recognizing the presented action.

DATA ACQUISITION CHALLENGES While the real data is representative of the environment at the RoboCup tournament, its collection is expensive and time-consuming. Besides the considerable annotation effort, setting up the field at different locations with diverse backgrounds and training individuals to perform required gestures adds to the expense and time commitment of data collection.

The real-time robotic framework deployed during data acquisition introduces additional challenges that can harm the data quality. For example, frame drops can happen and the camera can be reset after driver failures,

resulting in the sequence of frames obtained being non-consecutive. This leads to issues in synchronization and thereby, the frames need to be annotated manually, which increases annotation costs and can lead to additional sources of human error.

However, without such effort, the real data is not representative and cannot be used for training and testing without limitations. In our approach, we thus explore two different ways to approach this issue: generating fully synthetic data and recording chroma key sequences of real scenes with synthetic backgrounds for training purposes, reserving the real data exclusively for testing. These two different methods are explained in the next sections.

3.2.3 *Synthetic Data*

We create synthetic data by modeling the 3D simulation environment in the procedural 3D animation framework *Side FX Houdini*² that closely resembles the setup during RoboCup. Subsequently, photo-realistic referee action sequences are rendered from diverse camera views using a ray-tracing approach. Within the simulation environment, there is full flexibility to adjust camera positions, referee poses, referee body models, and textures. This facilitates the efficient creation of a diverse, large-scale dataset with precise and easy annotation of the generated video sequences.

SIMULATION ENVIRONMENT SETUP The simulation environment is set up by using the official field definition and a model of the NAO robot³ with differently colored jerseys as used during the real tournaments. To represent the referee, we employ rigged 3D human models that encompass different body shapes and textures. An example setup of our simulation environment is shown in Figure 3.2b.

Realistic and natural referee actions are generated from real videos that show the reference action as detailed in § 3.2.2. From these reference videos, we manually extract the position of human joints and subsequently apply these extracted joint movements to the rigged human model.

ROBOT POSITIONS In our simulated environment, the robots and cameras are randomly distributed across the field and remain in the same position when generating a single session of data. This provides the option

² <https://www.sidefx.com/>

³ <https://www.aldebaran.com/en/nao>

to fuse information from multiple robots that perform recognition in parallel. However, robots are randomly placed between each session to have the same variation as the real data.

We create a set of synchronized referee action sequences that encompass a wide range of viewpoints and game layouts. As cameras are distributed over the whole field, certain viewpoints are not suitable for observing the referee’s actions. This may be due to the location of a camera too close to the referee or acute angles relative to the referee’s orientation. In the former case, the referee’s hand can move out of the camera’s field of view, while in the latter scenario, strong ambiguity between the poses cannot be resolved even by a human annotator. To conduct a more detailed analysis of how the relative position between the robot camera and the referee impacts action recognition performance, we further categorize the camera positions. We label positions where the robots’ cameras are consistently a quarter of the field away from the referee and have a view angle of less than 45° angle as *easy positions*, and the other robot positions are labeled as *hard positions*. The synthesized robot positions are illustrated in Figure 3.4, with the easy position plotted in blue and the hard position indicated in red.

BACKGROUNDS Using the simulation environment, we render RGBA images with a transparent background that are further augmented with various backgrounds. We generate a total of 65 synthetic backgrounds with 53 used for training and 12 for validation. The backgrounds are generated using Stable Diffusion [193] trained on the 2b English language label subset of LAION 5b with prompts representative of the environments encountered during RoboCup such as crowded exhibition centers. An example of such a background with a real image (Section 3.2.4) in front is visible in Figure 3.3c.

3.2.4 Real Data - Chroma Key

A high-quality animation framework and raytracing renderer has been used to generate the synthetic data. However, they still do not represent images from the real images. We, therefore, collect more data from the NAO robots to bridge the gap between synthetic and real. Recording via a single robot generates only a single sample per location, which can either be used for training or validation. Thus, various sessions need to be generated at different locations to obtain diverse backgrounds. Green screen or chroma key backgrounds can allow multiple different backgrounds to be inserted after post-processing with an example provided in Figure 3.3. Thus, data

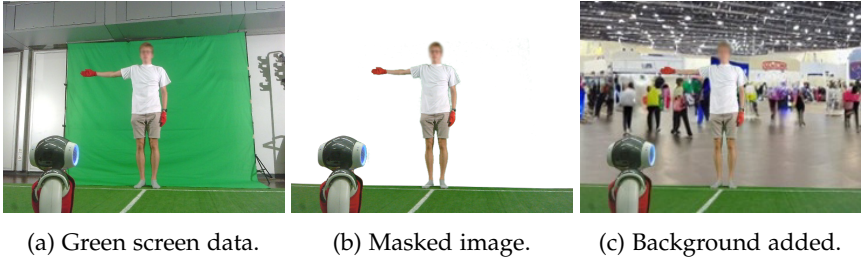


FIGURE 3.3: Data collection with green screen and augmentation with generated backgrounds.

can be recorded in a single setting and further images can be synthesized for different locations. The method also ensures synchronization across the data in different locations.

We use two different methods for data collection of chroma key images, which differ in the number of robots per session. In the first setting, referred to as Chroma Key Front (CK Front), a single robot is used. This robot is placed directly in front of the referee according to the official rules of RoboCup 2022. In total 9 people are participating in the data collection. In the second setting, referred to as Chroma Key Game (CK Game), multiple robots are used in the same session and placed in different positions on the field, following the rules of RoboCup 2023. In the second setting, 5 people participated in the data collection.

CHROMA KEY FRONT The first chroma key setting consists of videos acquired by a single robot placed in front of the referee. The referee is guaranteed to be fully in front of the greenscreen, which occupies the whole field of view. This ensures an easy extraction of the background. While the setting during recording is more limited and does not require operating multiple robots simultaneously, every sequence needs to be annotated manually. For each session, the greenscreen has been replaced with a transparent background by manually choosing a window of colors.

Following the RoboCup 2022 rules, class 12 is not present in this dataset. Therefore, this part of the dataset can be used to study the capabilities of our approach to learn with a data-mix where the class is only available in synthetic data. For future rule changes, this can indicate, how much new real data needs to be collected

CHROMA KEY GAME In the second setting called Chroma Key Game, all robots are randomly placed on the field with the referee in their field of view. This layout is changed for each session to provide sufficient variability. Figure 3.3a shows the view from one of the robots. The timestamps of the recorded frames are saved as the human performs the different gestures. Adobe Premiere has been used to generate a mask that removes the greenscreen. However, as the conditions are more diverse than in the first chroma key dataset, manual annotation of all frames is complex. The same methodology as for annotation of real data has been used, which helps to synchronize annotations between robots.

3.2.5 Data Split

The real data has been solely used for the purpose of testing. For the training and validation split, the backgrounds have been divided into training and validation backgrounds to avoid any information leakage. Furthermore, the people participating as referees in the dataset collection have been distributed into separate training and validation sets, such that there is no overlap in referees between the training and validation data.

The final dataset contains 183'591 images which are distributed accordingly: 69'323 synthetic, 43'699 chromakey game, 11'226 chromakey front and 59'451 real game. Figure 3.4 displays the locations on the field where the cameras are placed to simulate the synthetic environment. The locations are labeled as hard if they are less than 1.5m from the sideline where the referee is standing, or the angle at which the referee is seen is greater than 45°.

The classes in each of the domains are represented uniformly. However, due to following the RoboCup 2022 guidelines, class 12 does not exist in the CK front datasplit. In order to preserve the privacy of the different participants in the real datasets, their faces have been blurred.

3.3 ACTION RECOGNITION

To gain deep insight into our dataset and to provide a public benchmarking model to all RoboCup teams, we develop an approach for human action recognition designed for low-resource contexts. The method employs a MobileNet [100] architecture for image feature extraction as a backbone. After resizing each image from a window of 15 frames to 90 x 120 px, the corresponding deep feature is extracted. To further capture the temporal

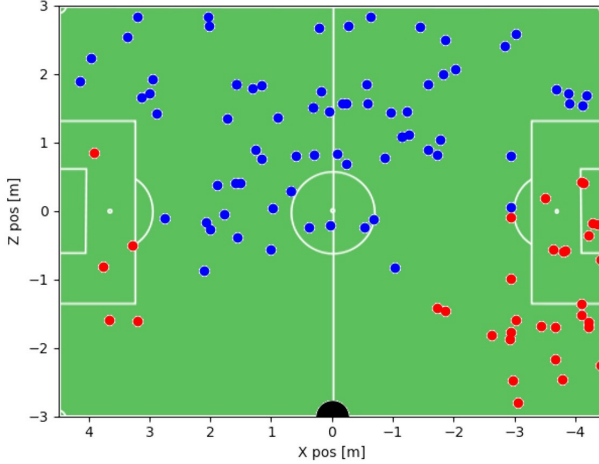


FIGURE 3.4: Synthetic data camera positions. Hard and Easy positions are represented by Red and Blue respectively. Referee coordinates are: $X = 0.0m$, $Z = -3.0m$.

relationships among the images, the sequence of 15 deep features is further processed by a gated recurrent unit GRU [44]. The GRU's 64-dimensional output is directed through 2 subsequent dense layers, each with a preceding Dropout layer [208] and ReLU [3] activation functions. Finally, the class is predicted directly from the logits. Our approach is further depicted in Figure 3.5 for clarity.

The model is trained end-to-end with an initial learning rate of $5e-4$ for 40 epochs and a plateau-based scheduler. Adam optimizer [123] is used to update all the trainable parameters in the model. Early Stopping [35] has been used to stop the model training if the validation loss does not improve for 15 epochs. The best model on the validation set is saved and used for evaluation on the test set.

For the training data, we aim for 60 frames for each action and video. As our model requires 15 frame sequences, the following steps are taken to sample the data. From the 60 frames in a sample, a window of 15 frames with a random starting point is chosen when training the model. For the test set, a fixed set of 15 frames starting at frame 10 of each 60-frame sample is selected for evaluating our model. All images are resized to 90×120 px, however we also provide the full image resolution in our dataset. For training our models at a higher speed, we further preprocess the training

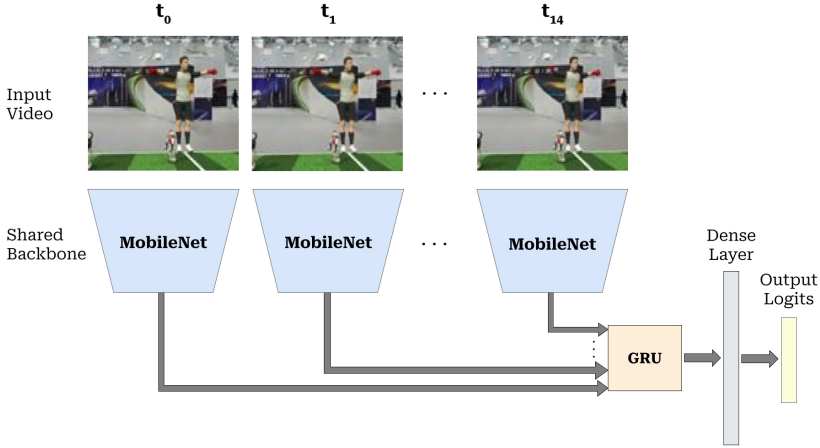


FIGURE 3.5: Overall Pipeline for Action Recognition.

data by randomly selecting 20 of the backgrounds for each sample and augmenting the images with them.

3.4 EXPERIMENTS AND RESULTS

RESULTS In the first set of experiments, we evaluate the basic training approach of using single-domain data for training to establish the baseline performance and to further investigate the usage of different combinations of synthetic, chroma key data. For all experiments, we split the synthetic training data into easy and hard sets as described in § 3.2.3. Furthermore, the evaluation is performed on real test data that is further separated manually into easy and hard examples as described in § 3.2.2. Considering the application of RoboCup, the test easy class is of major interest, as it best represents the current tournament scenario where only the robot locations that are known to have good viewing angles need to be considered for making a decision. In addition to evaluating the overall action classification accuracy, we take note of the fact that many actions possess mirror pairs (as illustrated in Figure 3.1). Consequently, we are also interested in assessing the accuracy of direction classification and gesture recognition. All evaluation results are provided in Table 3.1. In this section, the domains

synthetic and chroma key will be indicated by their abbreviations SYN and CK respectively.

SINGLE DOMAIN PERFORMANCE Investigating the set of single domain experiments in the first 5 rows of Table 3.1, the performance improves for an increasing overlap between the training and testing domains. On the full test set, this corresponds to the sequence of SYN, CK Front and CK Game. CK Game has the strongest performance by a large margin, with 60.4% and 65.0% accuracy on test full and easy respectively. SYN and CK front both exhibit a considerably lower performance, which can be attributed to the two different domain gaps. The former has a considerably different image appearance, while the latter covers a much smaller domain of viewing angles.

Further investigating training on SYN shows that the performance is strongly dependent on the location of the robots. The test performance degrades for all scenarios when utilizing SYN hard during training. This can likely be attributed to the observation that many actions cannot be easily recognized from hard positions, which results in the wrong signal being backpropagated to the model, degrading the performance.

MULTI-DOMAIN PERFORMANCE We evaluate the multi-domain performance by testing different combinations of SYN, CK Front, and CK Game during training. This can be used to make decisions about which kind of data needs to be collected when new rules or actions are introduced and how it can be augmented with synthetic data that can easily be adapted and regenerated. The results are presented in the second block of rows in Table 3.1.

Combining SYN and CK Front boosts the performance considerably, even though both datasets on their own have a large domain gap with the test real data. Compared with only training on SYN and CK Front, performance rises by 24.8% and 22.0% on test full respectively. This improvement can be explained by the complementary nature of the domain gaps which allows the training to cover the full domain when using them together. Further adding the CK Game data allows us to raise the model's accuracy to 76.1%, 85.6%, and 55.4% for test full, easy, and hard respectively. For our task, this supports the use of a multi-domain dataset, that contains large portions of data that are cheap to generate on a large scale.

Experiment	SYN easy	SYN hard	CK front	CK game	Test full	Test easy	Test hard	Gesture	Direction
single domain	✓				27.9	33.3	16.1	32.4	50.9
	✓	✓			21.9	25.2	14.6	26.6	57.5
			✓		30.8	28.0	37.1	33.8	42.8
				✓	60.4	65.0	50.2	62.6	77.3
			✓	✓	69.3	74.3	<u>58.4</u>	70.5	80.8
domain combination	✓		✓		52.8	54.7	48.7	55.5	70.5
	✓			✓	74.4	<u>82.5</u>	56.6	<u>74.7</u>	<u>83.3</u>
	✓		✓	✓	76.1	85.6	55.4	76.1	83.5
	✓	✓	✓		46.8	52.5	34.5	48.2	60.6
	✓	✓		✓	72.5	81.3	53.2	73.1	82.9
	✓	✓	✓	✓	<u>73.3</u>	79.4	59.9	74.0	84.7

TABLE 3.1: Evaluation on Test Set. Best performance is in bold and the second best has been underlined.

AMOUNT OF DATA As the data collection and annotation require a large amount of resources, we provide an analysis of the amount of data required to train the model. Thus, the model is trained on SYN together with only a subset of the referees available in the two CK datasets to investigate the influence on the model performance, which is depicted in Figure 3.6. The results indicate that even combining SYN data with a single referee from a CK dataset can improve the performance considerably, which is a promising perspective for data collection.

FEATURE ANALYSIS To gain insights into the learned multi-domain embedding space, we employ Uniform Manifold Approximation and Projection (UMAP) [161]. We visualize the image feature space for the model trained with SYN, CK Front, and CK Game data. The features from the datasets SYN, CK Front, CK Game and Test are depicted in Figure 3.7a and show more details for two class clusters in Figures 3.7b and 3.7c. The complete UMAP in Figure 3.7a indicates that different actions are well separated from each other. Investigating the two clusters in more detail

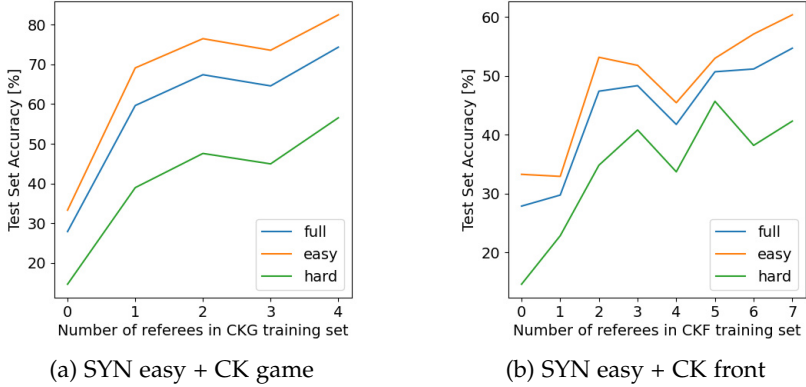


FIGURE 3.6: Test set performance when trained on varying amounts of real data.

shows that the real test data (red) covers the largest domain, with most features accumulating in one small location. SYN and CK Game cover slightly smaller parts of the cluster, both with a focus on different regions, but mostly contain Test. CK Front occupies the most minimal region that remains distinguishable from the Test cluster, indicating the presence of a domain gap. The findings in the multi-domain feature embedding analysis align with the experimental results in Table 3.1.

INTERPRETABILITY Saliency maps provide insight into the model’s decision-making process.. To understand how the model classifies different actions, we use RISE [181] to generate saliency maps. The method runs multiple predictions with different masks applied to the input image and relates them to the predictions on the masked images. The change in performance provides a heatmap that estimates the region where the model is focusing. As the method is originally intended to be used with single images, we adapted it to our dataset where the input to the model is 15 images. Based on the assumption that the referee motion is limited between frames, the same mask is used for all 15 images. 4000 masks are generated for each input and the initial masks are of size 16x16 which is then upsampled to the size of the image.

In Figure 3.8, we visualize the specific areas of interest where the model concentrates its attention. Notably, these regions align with the hand’s positions, which serve as the primary source of crucial information for the model to learn and recognize the performed actions.

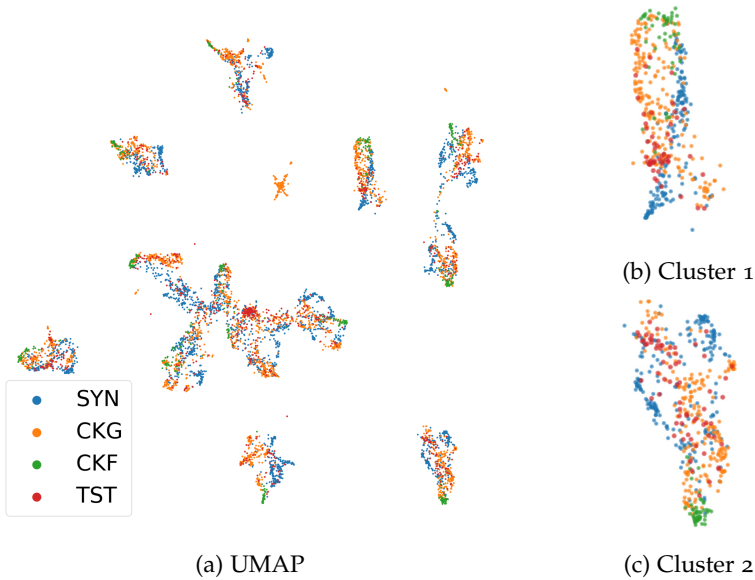
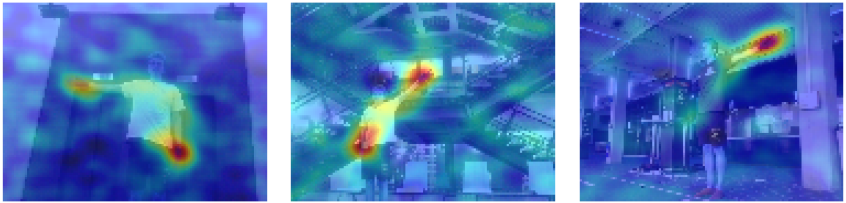


FIGURE 3.7: UMAP embedding of the different domains.

COMPUTATIONAL EFFICIENCY The model is converted to TFLite files so that we can efficiently deploy it on real robots with real-time inference. The feature extractor runs at 60 ms per image and when 15 images are obtained, the features are passed into the prediction model which generates the output at 4 ms per input.

3.5 CONCLUSION

In this chapter, we presented a new multi-domain referee action dataset that aims at providing the basis for bringing more autonomy to the RoboCup competition. Comprehensive experiments demonstrate that combining different domains improves the performance considerably and allows easy adaptability of the dataset to future rule changes. Finally, the implemented action recognition method is able to run real-time on low-performance robot hardware and can serve as a baseline to benchmark future approaches.



(a) Saliency for Pose 1. (b) Saliency for Pose 4. (c) Saliency for Pose 4.

FIGURE 3.8: Saliency Maps on Test Set.

Part II

MULTI-OBJECT TRACKING

Tracking is a core task in computer vision that forms the basis of dynamic world models to allow autonomous agents to understand their surroundings. Without any human input, the goal is thus to robustly follow all objects of a given set of relevant classes throughout the scene. While we as humans are able to perform this task easily on a limited number of objects, it already requires strong reasoning capabilities in scenarios where occlusions, appearance changes or fast movements are present. One key aspect is the ability to model object permanence during occlusions together with the objects' future position to re-identify them. We approach this task in Chapter 4, where 3D motion models are combined with a graph-neural network architecture to learn object association. Subsequently, Chapter 5 investigates the tracking of multiple identical robots, where re-identification becomes an even harder task. We solve this by fusing information from robot-mounted sensors with an external camera to perform tracking over long videos.

II.1 RELATED WORK

Multi Object Tracking (MOT) approaches the task of tracking all objects belonging to a given set of categories. These often include pedestrians in datasets containing videos from fixed surveillance cameras [53, 134, 155, 165]. Data collected from moving vehicles extends this with additional sensor modalities like LIDAR or radar and provides rich map information [31, 39, 76, 211]. These datasets also typically span a large number of different traffic participants like cars, buses, bicycles and pedestrians.

Existing tracking methods typically either approach 2D or 3D tracking, as each of the domains provides fundamentally different cues that can be exploited. While 3D methods profit from well-behaved motion models [43], 2D tracking can usually build on top of discriminative visual appearance features [90, 114, 141, 150, 176, 258]. For both 3D and 2D, trackers can be grouped into tracking by detection [20, 28, 95, 166] and joint tracking and detection methods [18, 162, 239]. Tracking by detection starts with a set of detected objects in every frame and links them subsequently [20], which is the most popular method used in 3D detection as of today. Joint tracking and detection builds on the perspective that detecting and tracking objects is a strongly interwoven process and thus, should be approached jointly [18, 239]. This approach is primarily chosen in 2D tracking, where object detectors are more mature than in the 3D domain and can be integrated well into the tracking pipeline.

2D MOT is well investigated, with the MOT challenge [53, 134, 165] and its corresponding datasets as the current performance reference.

A widely adopted approach to MOT is **tracking by detection**, where detections are available from an independently trained detection module and data association is performed by the tracker [28, 95, 166]. The association step is based on pairwise similarities between objects and aims to find a global solution to this problem. Similarities can contain purely geometric information of bounding boxes [20] or extend it with additional appearance features [233, 234]. With every object detector providing imperfect predictions, a core task of the assignment step also is the interpolation of occlusions and missed frames as well as the rejection of false-positive detections.

The data association step itself is closely linked to a hard-to-solve discrete optimization problem. This can be approached either by directly formulating and solving it or by implicitly representing the task using deep learning. Mapping the association step to a deep learning task [28, 48, 260] allows to further process similarity metrics, as well as to directly use large feature vectors that describe each object. However, simple heuristics are often required to resolve remaining inconsistencies that arise due to not being able to directly enforce constraints onto the network output. Using these approaches allows for training the complete pipeline end-to-end, without the direct requirement to define a cost for data association [28]. This work by Braso *et al.* [28] is also closest to our 3D-tracking approach presented in Chapter 4. It introduces Neural Message Passing (NMP) as a graph neural solver for offline 2D pedestrian tracking. Starting from a network flow formulation, the problem is transformed into a classification problem and data assignment is solved with an NMP network.

The alternative of explicitly stating an optimization problem [95, 96, 140, 191, 195, 214] requires a larger extend of modeling the task, but at the same time also allows for integrating prior information about the nature of tracks in an intuitive and transparent way. Due to the nature of the task, a wide range of approaches casts tracking as a graph problem [140, 195, 214] or network flow optimization [95, 96]. Nevertheless, these properties come at a high computational cost. As most of the proposed optimization problems are NP-hard [75], a considerable effort was invested in finding heuristics and approximate solvers for them [96].

Furthermore, as motion and appearance in 2D videos are partly predictable by simple models, Bayesian filtering has been employed for MOT by some earlier works. The filter is used to predict and estimate object

states, which also motivates our proposed pipeline. Important methods in this group are multi-hypothesis tracking [188], the Joint Probabilistic Data Association Filter [15], and PHD filters [79].

Following the paradigm of **joint detection and tracking**, where both steps are combined as a single module, Tracktor [18] uses the box regression module of faster RCNN [190] to propagate and refine object bounding boxes between frames. Related to this, Xu *et al.* [239] propose a differentiable approximation of Hungarian matching that allows end-to-end training of trackers. More recently, Meinhardt *et al.* [162] proposed a transformer architecture in the joint detection and tracking framework. A range of tracker extensions are commonly used in all approaches, including modules such as camera motion compensation [18] or object re-identification (ReID) [114, 150, 258]. In general, most of the 2D MOT methods profit from the high framerate available in videos [18]. Furthermore, state-of-the-art 2D object detectors achieve a high accuracy [87, 190, 213], such that the focus of tracking has shifted from the rejection of false positives towards a pure data assignment task [28].

The generation of **appearance features** and person **re-identification** is a core component of many tracking approaches, as it provides strong cues to match pedestrians after occlusions or crossing paths. A common paradigm in this context is metric learning [141, 176, 233], where features are learned together with a metric that measures the similarity between objects. This aims at jointly finding an embedding space and corresponding learned metric to distinguish between different pedestrians.

However, training data for re-identification raises strong privacy protection concerns which have recently led to a movement towards training the module on primarily synthetic data. In this context, PersonX [212] and Bak *et al.* [13] are notable examples that use a small set of models to generate training data. Pushing towards surpassing the scale of human-generated data, RandPerson [225] and ClonedPerson [224] further propose to automate this pipeline by generating randomized character clothing.

3D MOT extends the challenge of MOT to tracking multiple objects in 3D [31, 76]. With 3D MOT as a problem at the core of autonomous driving, a wide range of datasets that focus on tracking of objects in driving scenes is available [31, 39, 116, 211]. Due to the nature of the task, 3D MOT is usually performed online, which adds additional challenges and requires more heuristics. For detecting objects, any 3D modality would be suitable, nevertheless, most datasets provide LIDAR scans which are used in most tracking methods, including ours. As 3D object detection from LIDAR is

still an open research question and less robust than 2D detection, 3D MOT mostly follows the tracking by detection framework [43, 119, 229, 230, 244].

One line of work in 3D MOT establishes tracks directly from the output of an object detector and forms tracks by connecting detected objects between frames. These approaches can directly use the output of an object detector [244] or include more advanced features like additional 2D information [230, 254]. In this framework, Weng *et al.* [230] are the first to use a graph neural network to estimate the affinity matrix, which is then solved using the Hungarian algorithm. Since this group of trackers does not establish a predictive model for each track, they cannot directly account for missed detections or occlusions and require heuristics for these cases.

Another group of 3D trackers [43, 119, 229] resolves this issue by generating a separate representation of tracks and performs tracking by matching active tracks and detections at each timestep. AB₃DMOT [229] uses a Kalman filter [113] to represent the track state and matches tracks and detections based on intersection over union (IoU). Chiu *et al.* [43] extend this approach by matching based on the Mahalanobis distance [153] to resolve the issue that object size, orientation and position are on different scales. EagerMOT [119] uses tracks parameterized in 2D and 3D simultaneously to gain performance from multiple modalities. All of these approaches rely on heuristics to generate new tracks, as track initialization can hardly be learned in a purely offline training approach.

LEARNABLE ONLINE GRAPH REPRESENTATIONS FOR 3D MULTI-OBJECT TRACKING

Autonomous systems that operate in dynamic environments require robust object tracking in 3D as one of their key components. Most recent approaches for 3D MOT from LIDAR use object dynamics together with a set of handcrafted features to match detections of objects across multiple frames. However, manually designing such features and heuristics is cumbersome and often leads to suboptimal performance.

With our approach, we instead strive towards a unified and learning-based approach to the 3D MOT problem. We design a graph structure to jointly process detection and track states in an online manner. To this end, we employ a neural message-passing network for data association that is fully trainable. Our approach provides a natural way for track initialization and handling of false positive detections, while significantly improving track stability. We demonstrate the merit of the proposed approach in the nuScenes tracking challenge 2021 with a state-of-the-art performance of 65.6% AMOTA with 58% fewer ID-switches, resulting in the best LIDAR only submission and an overall second place.

4.1 INTRODUCTION

Autonomous systems require a comprehensive understanding of their environment for a safe and efficient operation. A task at the core of this problem is the capability to robustly track objects in 3D in an online-setting, which enables further downstream tasks like path-planning and trajectory prediction [6, 92, 248]. Nevertheless, tracking multiple objects in 3D in order to operate an autonomous system, poses major challenges. First, in the online setting, data association, track initialization, and termination need to be solved under additional uncertainty, as only past and current observations can be utilized. Furthermore, covering occlusions requires extrapolation with a predictive model rather than interpolation as in the offline case. Finally, when using LIDAR for data acquisition, no comprehensive appearance data is available and data association needs to primarily rely on object

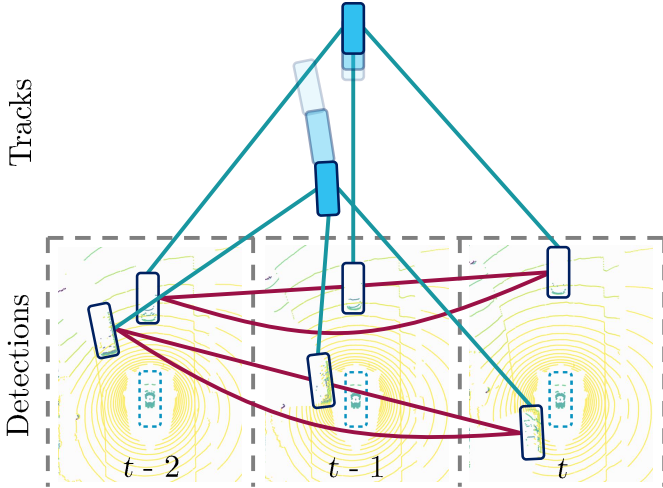


FIGURE 4.1: The proposed method uses a graph representation for detections and tracks. A neural message passing based architecture performs matching of detections and tracks and provides a learning based framework for track initialization, effectively replacing heuristics required in current methods.

dynamics. This is further complicated by the presence of fast moving objects such as cars.

With the release of large scale datasets for 3D tracking [31, 39, 116, 211], a considerable amount of work on 3D MOT has been initiated [43, 119, 229, 230, 244]. Most of these works address the aforementioned challenges by either linking detections directly in a learning based manner or use comprehensive motion models together with handcrafted matching metrics. All of these methods require a large set of heuristics and, to the best of our knowledge, none of the methods approaches the aforementioned challenges jointly. In contrast to this, recent work in 2D MOT [18, 28] aims to reduce the amount of heuristics by modeling all tasks in a single learnable pipeline using graph neural networks. However, most of these approaches are limited to the offline setting and driven by appearance-based association that cannot be readily employed in the 3D counterpart.

To establish the missing link between learning based methods and powerful predictive models in 3D MOT, we propose a unified graph representation that merges tracks and their predictive models with object detections into a single graph. This learnable formulation effectively replaces heuristics that

are required in current methods. A visualization of the graph is depicted in Figure 4.1.

Contrary to previous works, our learnable matching between tracks and detections is integrated into a closed-loop tracking pipeline, alleviating the need for handcrafted distance metrics. However, this raises the question of how to effectively train such a learnable system, as the generated tracks influence the data distribution seen during subsequent iterations. In this chapter, we propose and describe a two-stage training procedure for semi-online training of the algorithm, where the data seen during training is generated by the model itself. In summary, the contributions of our work are threefold:

- A unified graph representation for learnable online 3D MOT that jointly utilizes predictive models and object detection features.
- A track-detection association method that explicitly utilizes relational information between detections to further improve track stability.
- A training strategy that allows us to faithfully model online inference during learning itself.

We perform extensive experiments on the challenging nuScenes dataset. Our approach sets a new state-of-the-art, achieving an AMOTA score of 0.656 while reducing the number of ID-switches by 58%.

4.2 METHOD

We model the online 3D MOT problem on a graph, where detections are nodes and the optimal sequences of edges that connect the same objects throughout time need to be found. The resulting core tasks are data association by matching of nodes, track initialization while rejecting false positive detections, interpolation of missed/occluded detections, and termination of old tracks.

Without access to future frames due to the time causal nature in the online setting, all of the aforementioned tasks become challenging. In the case of track initialization, for instance, a new detection in the current frame with no link to a track could be a false positive or the first detection of a new track. And similarly for track termination, where an existing track that is not matched to any detection in the current frame may need to be terminated or may only encounter a missed or occluded object. While these dilemmas could often be resolved when future frames become available over time, online tracking performance is crucial for real-time decision systems since it directly influences the behavior of the system.

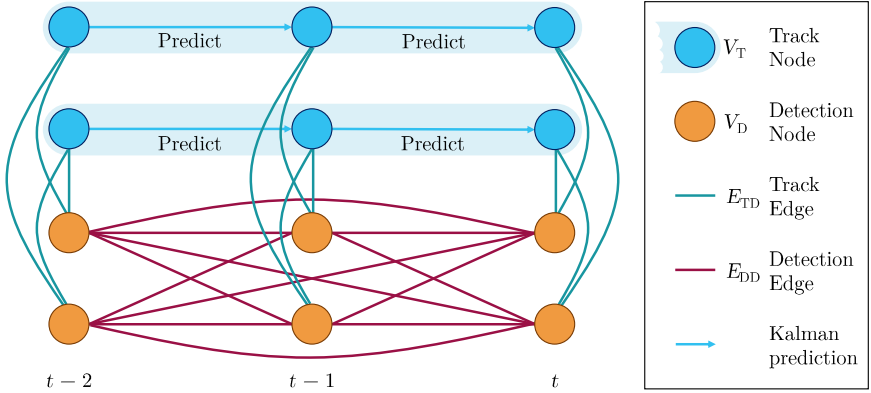


FIGURE 4.2: The proposed tracking graph combines tracks, represented by a sequence of track nodes and detections in a single representation. During the NMP iterations, information is exchanged between nodes and edges, and thus, distributed globally throughout the graph.

To jointly resolve these challenges in a learnable framework, we formulate a graph that merges tracks with their underlying dynamic model and detections into a single representation for online MOT. Based on the detections of the last T frames and the active tracks, a graph is built that represents the possible connections between tracks and detections. Starting with local features at every node and edge, NMP is used to distribute information through the graph and to merge it with the local information at each edge and node during multiple iterations. Finally, edges and nodes are classified as active or inactive. Based on the active edges that connect track and detection nodes, we formulate an optimization problem for data association. This jointly considers matches between tracks and detections and matches between detections at different timesteps to improve the track stability. Based on the connectivity of the remaining active detection nodes, tracks are initialized.

4.2.1 Graph Representation of Online MOT

Approaching 3D MOT as tracking by detection can be formulated as finding the set of tracks $\mathcal{T} = \{T_1, \dots, T_m\}$ that underlie the observed set of noisy detections. We parameterize a track as the state of the underlying Kalman

filter and a detection by its estimated parameters such as bounding box, class and velocity. To find a robust and time-consistent solution, three tasks need to be solved:

1. Assignment of detections to existing track.
2. Linking of detections across timesteps.
3. Classification of false positive detections.

While either 1. and 2. would be sufficient on their own to perform tracking, finding a joint solution promotes stability of the tracks. Furthermore, utilizing a track model is beneficial, since it aggregates information of the complete sequence of matched observations which is required to interpolate missing detections.

The three tracking tasks can be naturally formulated as one joint classification problem on a tracking graph $G = (V_D, V_T, E_{DD}, E_{TD})$. The graph is built from detection nodes V_D , track nodes V_T , detection edges E_{DD} that connect pairs of detection nodes at different timesteps and track edges E_{TD} that connect track and detection nodes at the same timestep. The complete tracking graph is visualized in Figure 4.2. Note that track nodes have sparser connections than detection nodes. They are only connected to the detections at the same timestep and to the neighboring timesteps of the same track. We chose this pattern since connected tracks and detections need to be temporally consistent and the relation between track nodes is determined by the Kalman prediction step. One additional characteristic of track nodes is that nodes that correspond to the same track form a track-subgraph called $G_{T,m}$, which is highlighted with a blue shaded area in Figure 4.2. These subgraphs are important since they share the same state that is linked with a dynamic model. Next, we discuss the types of nodes and edges used in our graph in more detail.

Notation: Symbols with subscript D belong to detection nodes and symbols with subscript T to track nodes. Symbols with subscript DD belong to detection edges and symbols with subscript TD to track edges.

Nodes are indexed with integer numbers from the set \mathcal{I} for detection nodes and from \mathcal{K} for track nodes. Edges are referred to by the indices of the connected nodes, i.e. $E_{TD,ki}$ describes a track edge from $V_{T,k}$ to $V_{D,i}$. As the graph is undirected, the notation also holds when the order of the indices is switched. To make our notation easy to read we always use the same index variables. More precisely, the index variables $i, j, m \in \mathcal{I}$ are used to refer to detection node indices and index variables $k, p, q \in \mathcal{K}$ refer to track node indices. The newest timeframe available to the algorithm

during online tracking is denoted as t and the timeframe of a specific node is referred to as t_i . Finally, tracks are indexed with their track ID n .

Detection nodes are generated for the detected objects and are initialized from the feature $\mathbf{x}_{D,i}$ containing the position, size, velocity, orientation, one-hot encoded class, detection score, and the distance of the detected object relative to the acquisition vehicle. The position is given in a unified coordinate system that is centered at the mean of the detections in the graph. The orientation, relative to the same unified coordinate system, is expressed by the angle's sin and cos.

Track nodes represent the state of an active track, i.e., each track generates one track node at every timestep. This groups the track nodes into track-subgraphs. The feature $\mathbf{x}_{T,k}$ at every track node is defined by the position, size, orientation, and the one-hot encoded class of the tracked object. The tracks are modeled by a Kalman filter with 11 states corresponding to the position, orientation, size, velocity and angular velocity. Parameters are learned from the training set as proposed by [43].

Detection edges refer to edges between a pair of detection nodes $V_{D,i}, V_{D,j}$ at two different frames $t_i \neq t_j$. They are parameterized by $\mathbf{x}_{DD,ij}$ containing the frame time difference, position difference, size difference, and the differences in the predicted position assuming constant velocity. To reduce the connectivity of the tracking graph, detection edges are only established between detections of the same class and truncated with a threshold on the maximal distance between two nodes. This implicitly corresponds to a constraint on the maximum velocity an object can achieve. Graph truncation makes inference more efficient, track sampling more robust and helps to reduce the strong data imbalance between active and inactive edges.

Track edges are connections between a track node $V_{T,k}$ and a detection node $V_{D,i}$ at the same timestep $t_k = t_i$. These edges are modeled with the feature $\mathbf{x}_{TD,ki}$, containing the differences in position, size and rotation.

Classification Given the unified graph G , the tracking problem is transformed to the following classification tasks:

1. Classification of active track edges E_{TD} .
2. Classification of active detection edges E_{DD} .
3. Classification of active detection nodes V_D .

Our approach to solving these tasks jointly is presented in the following.

4.2.2 Neural Message Passing for Online Tracking

Given only the raw information described in the previous section, classifying edges as active is hard and error-prone. To generate a good assignment, the network should have access to the global and local information present in the tracking graph. To archive this exchange of information within the graph, we rely on a graph-NMP network. We follow the notation of NMP used in [28] as some parts of the NMP processing are shared with this work and highlight similarities and differences. In the following we present the four stages of our algorithm:

1) Feature embedding: The input to the NMP network are embeddings of the raw edge and node features. To generate the 128 dimensional embeddings, the raw features are normalized and subsequently processed with one of four different Multi-Layer Perceptrons (MLP), one for each type of node/edge. This results in the initial features $h_{D,i}^{(0)}, h_{T,k}^{(0)}, h_{DD,ij}^{(0)}, h_{TD,ki}^{(0)}$.

2) Neural message passing: Initially, all information contained in the embeddings is local and thus, not sufficient for directly solving the data assignment problem. Therefore, the initial embeddings are updated using multiple iterations of NMP that distribute information throughout the graph. An NMP iteration consists of two steps. First, the edges of the graph are updated based on the features of the connected nodes. In the second step, the features of the nodes are updated based on the features of the connected edges. The networks used to process messages in NMP are shared between all iterations $l = 1, \dots, L$ of the algorithm. Next, we will describe the NMP iteration for each node and edge type in detail.

Detection Edges $E_{DD,ij}$ at iteration l are updated with a single MLP \mathcal{N}_{DD} that takes as an input the features of the two connected detection nodes $h_{D,i}^{(l-1)}, h_{D,j}^{(l-1)}$, the current feature of the edge $h_{DD,ij}^{(l-1)}$ and the initial feature $h_{DD,ij}^{(0)}$

$$h_{DD,ij}^{(l)} = \mathcal{N}_{DD} \left([h_{D,i}^{(l-1)}, h_{D,j}^{(l-1)}, h_{DD,ij}^{(l-1)}, h_{DD,ij}^{(0)}] \right). \quad (4.1)$$

We add the current and initial edge feature to the input vector, introducing a skip connection into the unrolled algorithm.

Track edges $E_{TD,ki}$ are updated according to the same principle as detection edges, using information from connected nodes, but with a separately trained MLP \mathcal{N}_{TD} . The update rule is given as

$$h_{TD,ki}^{(l)} = \mathcal{N}_{TD} \left([h_{T,k}^{(l-1)}, h_{D,i}^{(l-1)}, h_{TD,ki}^{(l-1)}, h_{TD,ki}^{(0)}] \right). \quad (4.2)$$

Note that this type of edge is new to our formulation since we introduced track nodes.

Detection nodes are updated with a time-aware node model in [28]. We further process the additional input from the track edges, introduced in our formulation, using a similar time-aware model. Given a fixed detection node $V_{D,i}$, messages are generated for every detection edge $E_{DD,ij}$ and tracking edge $E_{TD,ki}$ connected to it. To handle detection edges to future frames, to past frames and track edges separately, three MLPs $\mathcal{N}_D \in \{\mathcal{N}_D^{\text{past}}, \mathcal{N}_D^{\text{fut}}, \mathcal{N}_D^{\text{track}}\}$ are employed for this task, using the following prototype function

$$m_{D,ij}^{(l)} = \mathcal{N}_D \left([h_{DD/TD,ij}^{(l)}, h_{D,i}^{(l-1)}, h_{D,i}^{(0)}] \right). \quad (4.3)$$

All networks get the current and initial feature of node $V_{D,i}$ as an input to establish skip connections. Note that in the first and last time frame, where no past respectively future edges are available, zero padding is used.

The messages formed at the incident nodes are aggregated separately for the three types of connections by a symmetric aggregation function Φ , which is the summation aggregation function in our implementation. The node feature is updated with the output of a linear layer, processing the aggregated messages as

$$h_{D,i}^{(l)} = \mathcal{N}_D \left([m_{DD,i,past}^{(l)}, m_{DD,i,fut}^{(l)}, m_{TD,i,track}^{(l)}] \right). \quad (4.4)$$

Track nodes are new in our pipeline and require different processing than detection nodes. As messages are sent from track edges only at the same timeframe, the messages can be formed as

$$m_{T,ki}^{(l)} = \mathcal{N}_T \left([h_{TD,ki}^{(l)}, h_{T,k}^{(l-1)}, h_{T,k}^{(0)}] \right), \quad (4.5)$$

and accumulated using the aggregation function Φ as before

$$m_{T,k}^{(l)} = \Phi \left(\left\{ m_{T,ki}^{(l)} \right\}_{i \in N_k} \right). \quad (4.6)$$

Finally, the message is processed by a single linear layer

$$h_{T,k}^{(l)} = \mathcal{N}'_T \left(m_{T,k}^{(l)} \right). \quad (4.7)$$

These NMP steps are performed for L iterations, which generates a combination of local and global information at every node and edge of the graph.

3) Classification: The node and edge features available after performing NMP can be used to classify detection nodes, detection edges, and track edges as active or inactive. Detection nodes need to be classified as active if they are part of a track or initialize a new track and as inactive if they represent a false positive detection. Detection edges and track edges are classified as active if the adjacent nodes represent the same object. For each of the tasks, a separate MLP that takes the final features, $h_{D,i}^{(L)}$, $h_{DD,ij}^{(L)}$, and $h_{TD,ij}^{(L)}$, is used to estimate the labels $y_{D,i}$, $y_{DD,ij}$, and $y_{TD,ki}$. The result of the classification stage are three sets. First, the set of active detection node indices

$$\mathcal{A}_D = \{i \in \mathcal{I} \mid y_{D,i} \geq 0.5\}. \quad (4.8)$$

Secondly, the set of active detection edge indices

$$\mathcal{A}_{DD} = \{i, j \in \mathcal{I} \times \mathcal{I} \mid y_{DD,ij} \geq 0.5\}. \quad (4.9)$$

Finally, the set of active track edge indices

$$\mathcal{A}_{TD} = \{k, i \in \mathcal{K} \times \mathcal{I} \mid t_k = t_i \wedge y_{TD,ki} \geq 0.5\}. \quad (4.10)$$

While in [28] only \mathcal{A}_D is predicted, we infer the two additional sets to improve track stability (see Section 4.3).

Note that during training, classification is not only performed on the final features $h^{(L)}$ but also during earlier NMP iterations. This distributes the gradient information more evenly throughout the network and helps to reduce the risk of vanishing gradients.

4) Track update: In the last stage of our algorithm, we use the sets of active nodes and edges, to update and terminate existing tracks as well as to initialize new tracks. We achieve this with a greedy approach that maximizes the connectivity of the graph.

Updates of tracks are performed by finding the matching detection nodes in the graph for each track and time step. This is represented as an assignment, which is a set of detection node indices

$$\mathcal{F}_n \subset \mathcal{I} : |\mathcal{F}_n| \leq T \text{ and } \forall i, j \in \mathcal{F}_n : t_i \neq t_j \text{ if } i \neq j \quad (4.11)$$

from different timesteps. We define the best assignment as the set of indices corresponding to detection nodes that are 1) all connected to the track-subgraph $G_{T,n}$ and 2) have the most active detection edges connecting them with each other. To find the best assignment for a track n , we start with

the set of detection node indices that are connected to a track node $V_{T,k}$ through an active track edge.

$$\mathcal{C}_{D,k}^{node} = \{i \in \mathcal{I} \mid ki \in \mathcal{A}_{TD}\}. \quad (4.12)$$

By considering all track nodes of the track-subgraph $G_{T,n}$, the set of detection edge indices connected to a track is defined as

$$\mathcal{C}_{D,n} = \bigcup_{k \in G_{T,n}} \mathcal{C}_{D,k}^{node}. \quad (4.13)$$

Finally, the set of active detection edge indices between these nodes is derived as

$$\mathcal{C}_{DD,n} = \{ij \in \mathcal{C}_{D,n} \times \mathcal{C}_{D,n} \mid ij \in \mathcal{A}_{DD}\}. \quad (4.14)$$

The quality of the assignment Γ representing the optimization problem is the number of detection edges between the assignment nodes that is also present in $\mathcal{C}_{DD,n}$

$$\Gamma = |\{\mathcal{F}_n \times \mathcal{F}_n\} \cap \mathcal{C}_{DD,n}|. \quad (4.15)$$

A solution for all tracks is searched with a greedy algorithm, while never assigning a detection node multiple times. As older tracks are more likely true positive tracks, updating is done by descending age of tracks. If there are multiple solutions with the same cost, we employ the following tie breaking rules. First, solutions with the lowest number of nodes are selected. If this does not make the problem unambiguous, the solution that maximizes the sum of 3D detection scores of the selected detection nodes is chosen. A visualization of this approach is shown in Figure 4.3.

Termination of tracks is based on the time since the last update. If a track has not been updated for three timesteps or 1.5s, it is terminated.

Initialization of tracks takes into account detection nodes and the corresponding detection edges. Our approach consists of two steps, split over two consecutive frames. First, all active detection nodes in the most recent frame that have not been used for a track update are labeled as preliminary tracks. In the next iteration of the complete algorithm, these nodes are in the second to last frame. A full track is generated for each of these nodes that are connected to an unused active detection node in the newest frame by an active detection edge. If multiple active detection edges exist, the edge that connects to the node with the highest detection score is chosen.

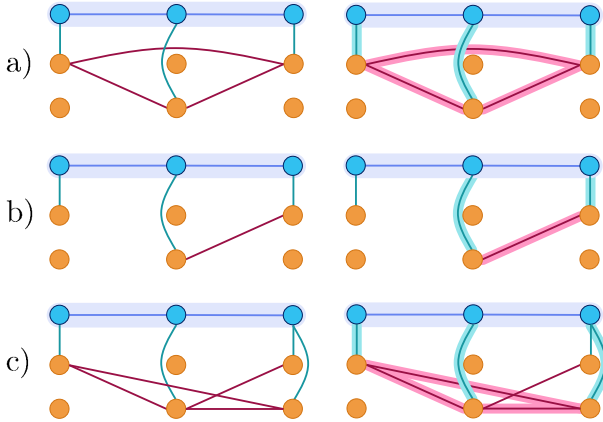


FIGURE 4.3: Visualization of different update scenarios, with only active edges in the graph. The graph represents a single track and two detections at each time step. a) Shows the ideal case where a track is matched to one node at every timestep and each detection node is connected with each other. b) Represents the case where a match at one timestep is dropped and the track is only matched to two detection nodes. c) Shows a situation, where the proposed approach is able to decide for the globally best solution, even though two detection nodes have been matched to the track in the last frame.

4.2.3 Training Approach

When training an online tracker, we face one fundamental challenge, which is the distribution mismatch of track nodes during training and inference. While the track nodes available during training are derived from the ground truth annotations in the dataset, the track nodes encountered during inference are generated by the algorithm itself in a closed loop.

Data augmentation: We use data augmentation to make the model more robust against changes in the distribution of tracks and detections as well as to simulate rare scenarios. Although the data naturally contains imperfections such as missed detections and noise on the physical properties of objects, we perform four additional data augmentation steps. Detections are dropped randomly from the graph to simulate missed or occluded detections. Noise is added to the position of the detected objects. This allows us to counteract the well-known issue of detector overfitting [31], where the detections used for training the tracking algorithm are considerably better than the detections available during inference, as the detector was trained

on the same data as the tracker. To model track termination, all detections assigned to randomly drawn tracks are removed. Finally, track initialization is simulated by dropping a complete track while keeping the corresponding detection nodes. This ensures that the case of track initialization is encountered often during training.

Two-stage training: Data augmentation helps to train a better data association model, however, even with data augmentation, the model does not learn to perform association decisions in a closed loop. To overcome this challenge one could train with fixed length episodes where only the beginning is determined by the ground truth. However, such an approach comes with two issues. First, it is inherently hard to train due to potentially large errors and exploding gradients. Secondly, this approach is computationally costly on large datasets as no precomputed data can be used. Thus, we propose a two-stage training scheme as an alternative that approaches the same challenge. In this setting, a model is trained first on offline data with strong data augmentation. To do so, the results obtained from a LIDAR detection model [244, 261] are matched with the annotation data available for the training and validation dataset. The detections matched to tracks are then processed with the Kalman filter model to generate track data for training.

After training the full model on the offline data with data augmentation, the model can be used for inference in an online setting. We run the tracker on the complete training dataset and generate tracks that show a distribution closer to the online-case. This results in a new dataset, which contains the same set of detections as before, but updated tracks. By retraining the model on this second stage dataset, together with all data augmentation steps used before, considerable performance gains can be accomplished.

Training parameters: We train all models with the Adam [123] optimizer for four epochs with a batch size of 16 and a learning rate of 0.0005. Focal loss [143] with $\beta = 1$ is used for classification of edges and nodes, weight decay is set to 0.01 and weights are initialized randomly. The MLPs used for embedding, NMP, and classification have [64, 128], [256, 256, 128], and [128, 32, 8] neurons in their respective layers. In all experiments, graphs with $T = 3$ timesteps are considered.

4.3 EXPERIMENTS AND RESULTS

All experiments are performed on the publicly available nuScenes dataset [31] with LIDAR detections only. Scores on the test set are centrally evaluated

Method	Detections	Data	AMOTA \uparrow	AMOTP \downarrow	MOTA \uparrow	MOTP \downarrow	IDS \downarrow	FRAG \downarrow
AB ₃ DMOT [229]	MEGVII [261]	3D	0.151	1.501 m	0.154	0.402 m	9027	2557
StanfordIPRL [43]	MEGVII [261]	3D	0.550	0.798 m	0.459	0.353 m	950	776
GNN ₃ DMOT* [229]	-	2D + 3D	0.298	-	0.235	-	-	-
CenterPoint [244]	CenterPoint [244]	3D	0.638	0.555 m	0.537	0.284 m	760	529
CenterPoint-Ensemble*	CenterPoint Ensemble*	3D	0.650	0.535 m	0.536	0.294 m	684	553
Ours	CenterPoint [244]	3D	0.656	0.620 m	0.554	0.303 m	288	371

TABLE 4.1: Results on the nuScenes test set. Methods marked with asterisk use private detections and thus, no direct comparison is possible. Benchmark available at nusenes.org/tracking with our method listed as *OGR₃MOT*.

and results on the validation set are computed with the official developer’s kit. NuScenes is known to be more challenging than previous datasets [230] and has a leaderboard with a range of current LIDAR based methods, thus, providing a suitable platform to test state-of-the-art detection and tracking approaches. To demonstrate that our method generalizes across significantly different object detectors and provides the same advantages in all scenarios, we perform all experiments with two different object detectors.

Detection Data. To verify the performance of our method with multiple detectors, we choose the two state-of-the-art detectors CenterPoint [244] and MEGVII [261] that are based on very different techniques. While CenterPoint currently provides the best performance of all publicly available methods, MEGVII is used by many previous methods. We perform all experiments with both detectors and thus, allow for a fair comparison between approaches.

MPN Baseline. To show the merit of an explicit graph representation, we implement our method without track nodes and track edges as a baseline. This corresponds to the direct adaptation of the tracker introduced in [28] to the online and 3D MOT setting. In this case, tracks are modeled as a sequence of detections and matching is performed with the classified detection edges and nodes. This method is denoted as MPN-baseline in the following.

Tracking Results. The results on the nuScenes test set are shown in Table 4.1. It depicts all competitive LIDAR based methods, which were benchmarked on nuScenes and have at least a preprint available. Our approach achieves an AMOTA score of 0.656, outperforming the state-of-the-art tracker CenterPoint [244] by 1.8% using the same set of detections. Compared to CenterPoint-Ensemble, which uses multiple models and an improved set of object detections that are not publicly available, we improve by 0.6%. Finally,

Method	AMOTAAMOTPMOTA IDS FRAG				
Detections: MEGVII [261]					
AB3DMOT [229]	0.509	0.994 m	0.453	1138	742
StanfordIPRL [43]	0.561	0.800 m	0.483	679	606
CenterPoint [244]	0.598	0.682 m	0.504	462	462
MPN-baseline	0.514	0.979 m	0.451	1389	520
Ours	0.631	0.762 m	0.541	263	305
Detections: CenterPoint [244]					
AB3DMOT [229]	0.578	0.807 m	0.514	1275	682
StanfordIPRL [43]	0.617	0.984 m	0.533	680	515
CenterPoint [244]	0.665	0.567 m	0.562	562	424
MPN-baseline	0.593	0.832 m	0.514	1079	474
Ours	0.693	0.627 m	0.602	262	332

TABLE 4.2: Results on the nuScenes validation set. MPN-baseline[†] corresponds to the method in [28] adapted to the online setting as described in Section 4.3.

ID switches and track fragmentation are reduced by 58% and 30% respectively. This improved track stability can be explained by the integration of the predictive track model into the learning framework.

Our algorithm runs with 12.3 fps or 81.3 ms latency on average on an Nvidia TitanXp GPU. As 57.8 ms of this time is used for graph generation and post-processing and only 23.4 ms is required for NMP and classification, major gains may be achieved with a more efficient implementation.

Table 4.2 shows the results of the current state-of-the-art 3D trackers with two different sets of detections, making them comparable. In this scenario, our approach gains 2.8% AMOTA score compared to CenterPoint [244] on their own detection data and 3.3% on the reference MEGVII [261] detections. Again the advantages of using a dedicated model for tracks becomes apparent in the number of ID-switches, which are reduced by 47% and 43% using our model on Centerpoint [244] and MEGVII [261], respectively.

Ablation Study. We evaluate the modules of our tracker in an ablation study shown in Table 4.3. We perform the full study on both sets of detections and for the two training scenarios. The results labeled *online* in Table 4.3 refer to our two-stage training pipeline and results labeled *offline* correspond to only training in the first stage of this approach, where no

data is generated by the tracker itself. In all cases, inference is performed online.

The results indicate that all implemented modules benefit our method. The highest impact is achieved by propagating information globally using NMP. Next to this, removing information from edges impacts performance for both training approaches. Without node information, the performance drop depends on the detector. While for Centerpoint the performance drop is severe, it is smaller in the case of MEGVII detections, especially in the offline training case. This may be explained by the quality of detections. While the position information is encoded on nodes and edges, information like object size is only contained on the nodes. Such information has only small variations between different objects and thus, it can only be used effectively if the detection quality is high, as given for CenterPoint.

To remove track nodes, we use the baseline implementation as introduced in Section 4.3. As only detections are used, this approach does not suffer from a distribution mismatch and two-stage training is neither necessary nor possible. Therefore, while the impact for offline training seems reasonable, the overall impact in the full method is significant. To show the benefit of using detection and track edges jointly for the track update, a naive matching only using track edges in the latest frame is used. This approach performs worse than not using a separate track representation at all and supports our approach of using global information for matching. Finally, focal loss gives a small advantage in all settings and data augmentation helps, especially for offline training. This can be explained, as in the two-stage training, the data distribution is closer to the distribution encountered during inference and thus, less data augmentation is required.

To further investigate our track termination approach, we investigate different track lifetimes in Table 4.4. Reducing the track lifetime leads to a sub-optimal performance for both AMOTA and ID switches, while an increased track lifetime improves the number of ID switches, but comes at the cost of a reduced AMOTA score. For different numbers of layers of the MLPs used during NMP, results are shown in Table 4.5. While adding an additional layer for a total of 4 layers reduces the number of ID switches, the corresponding AMOTA score is reduced. For a 5 layer MLP, the network does not converge which results in a severe performance reduction.

Method	CenterPoint [244]		MEGVII [261]	
	online	offline	online	offline
w/o NMP	0.427	0.427	0.557	0.405
w/o edge features	0.502	0.521	0.460	0.359
w/o node features	0.652	0.587	0.610	0.582
w/o track nodes	(0.593)	0.593	(0.513)	0.513
w/o det edges	0.544	0.482	0.487	0.423
naïve matching	0.607	0.576	0.529	0.500
w/o focal loss	0.684	0.647	0.618	0.581
w/o data augmentation	0.688	0.601	0.630	0.538
full pipeline	0.693	0.653	0.631	0.587

TABLE 4.3: Comparative ablation study performed with detections from CenterPoint [244] and MEGVII [261]. Numbers are AMOTA scores where online refers to the two-stage training introduced in Section 4.2.3 and offline to the basic training not using self-generated data.

Lifetime	1	2	3	4	5	MLP layers	2	3	4	5
MOTA	0.657	0.683	0.693	0.686	0.681	MOTA	0.682	0.693	0.681	0.468
IDS	581	345	262	260	226	IDS	297	262	243	12956

TABLE 4.4: Comparison of performance w.r.t. track lifetime for CenterPoint detections [244].

TABLE 4.5: Comparison of performance w.r.t. MLP depth for CenterPoint detections [244].

4.4 CONCLUSION

We proposed a unified tracking graph representation that combines detections and tracks in one graph, which improves tracking performance and replaces heuristics. We formulated the online tracking tasks as classification problems on the graph and solve them using NMP. To efficiently update tracks, we introduce a method that jointly utilizes matches between all types of nodes. For training, we propose a semi-online training approach that allows us to efficiently train the network for the closed-loop tracking task. Finally, we performed exhaustive numerical studies showing state-of-the-art performance with a drastically reduced number of ID switches. As our proposed method provides a flexible learning based framework, it

allows for a wide range of possible extensions and enables the way towards integrating fully learning based track state representations.

OPTIMIZING LONG-TERM ROBOT TRACKING WITH MULTI-PLATFORM SENSOR FUSION

Monitoring a fleet of robots requires stable long-term tracking with re-identification, which is yet an unsolved challenge in many scenarios. One application of this is the analysis of autonomous robotic soccer games at RoboCup. Tracking in these games requires the handling of identically looking players, strong occlusions, and non-professional video recordings, but also offers state information estimated by the robots. In order to make effective use of the information coming from the robot sensors, we propose a robust tracking and identification pipeline. It fuses external non-calibrated camera data with the robots' internal states using quadratic optimization for tracklet matching. The approach is validated using game recordings from previous RoboCup World Cup tournaments.

5.1 INTRODUCTION

Robust tracking with stable object identification is a crucial component in many robot applications. Previous works in related tasks use robot-mounted sensors in order to achieve the task in various settings [27, 32, 64, 199]. A related task in a different setting is analyzing motions of dynamic agents (e.g. humans in sports) through a fixed external camera [146, 147, 220, 221]. In our framework, we propose to fuse information from both types of sensors to robustly track humanoid robots in entire soccer game videos. Although the problem is closely related to automated game analytics, the availability and use of internal robot sensors brings its own unique applications, challenges and opportunities.

We focus on matches in the RoboCup Standard Platform League (SPL), where humanoid NAO robots from two teams compete fully autonomously in soccer matches. Robocup is an international annual competition where teams program different robots to compete in soccer. The long term goal of the project is to have a team of humanoid robots that can win against the winners of the World Cup in compliance with the official rules of FIFA. One of the main platforms in the competition is the SPL where two teams score using five NAO robots each. The actions performed by the robots

are autonomous and the first team to score 10 goals or the team with the highest number of goals after twenty minutes are announced the winners. The teams can only make changes to the software present in the NAO robots and no modifications to the hardware are allowed. Making game analytics available in this league can help teams improve their gameplay by providing an objective way of comparing the performance of their algorithms.

Our problem differs in multiple ways from the well-known tracking and identification problem game analytics: RoboCup games are recorded with non-professional uncalibrated camera equipment, robots look identical except for their jerseys, jersey numbers are too small to detect reliably, and human referees often occlude a significant part of the scene. These specifics introduce unique and non-trivial challenges into our long-term tracking task. In particular, the re-identification by recognition becomes virtually infeasible which is not the case in standard game analytics.

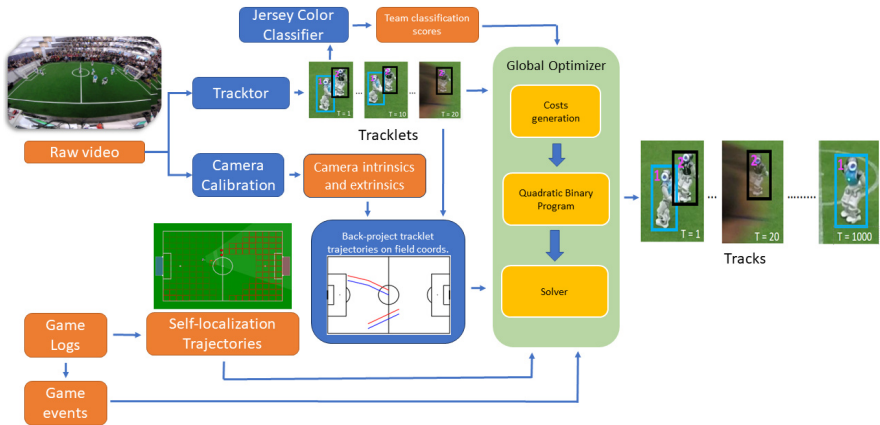


FIGURE 5.1: Overview of the proposed approach. The pipeline includes the processed raw video as well as the robot states as the inputs. The processed raw video provides the tracklets from Tracktor and the jersey/team classification as inputs to the optimizer. The robot states used as inputs are the self-localization and fallen state. Another important component that facilitates fusion of these inputs is the camera calibration module. The multi-modal inputs are fed into the global optimizer in order to generate the final track results.

Like previous methods, we start with the tracking of individual robots. Different tracking methods can be used on the video to generate tracklets. However, the tracklets are obtained solely from visual features and do not

extend to the whole duration of the game. We, therefore, opt for the use of the internal states of the robot in order to extract more useful attributes, which we call features. As we show in this chapter, these attributes can be efficiently used in order to match the different tracklets with the robot tracks. Therefore, we formulate the tracking problem as a biquadratic optimization where the internal states of the robots are used to provide different costs, used to collate the different tracklets. Overall, we propose a long-term tracking pipeline consisting of the following modules:

1. Camera calibration, to estimate camera intrinsics, including distortion, and the extrinsic camera pose relative to the playing field.
2. Short-term object tracking, to generate tracklets using Tracktor [18] with a Faster-RCNN [190] object detector pretrained on MS-COCO and finetuned on our dataset.
3. Long-term object tracking, to match tracklets to player identity by optimizing a quadratic problem, which fuses visual detections from the external camera and the robot's own self-localization and status messages.
4. Optimizing the long-term tracking performance by fine-tuning the weights associated with the cost terms.

GAME ANALYTICS One of the main problems in game analytics is tracking and identification of players in videos [149, 220, 240]. MOT is the first key component of the pipeline, which provides candidate detections of the players. Other components include team detection [240] or a combination of team and jersey identification [77]. The work by Maglo *et al.* [152] uses detection followed by association of tracklets in sports videos using player re-identification. In this case, tracklet association is also learned as the method does not have other inputs including spatial locations for the association. [146] on the other hand, uses the estimated spatial image locations of the players for the task. However, as our problem is different from standard game analytics formulations, the solutions presented in previous works [147, 152, 215, 220] are not directly applicable to our task. Specifically, a key problem that is not approached, is the full integration of the 3D environment as well as 3D localization of the players in player tracking and identification.

CAMERA CALIBRATION Exploiting the known 3D environment during tracking and identity assignment requires accurate camera intrinsics

and extrinsics, where identifying these parameters is performed by camera calibration. Standard calibration processes generally provide accurate intrinsic parameters [256] using multiple views of a calibration pattern. Alternative approaches without calibration patterns use minimal point correspondences [185] or a robot’s known motion for camera calibration [186] by evaluating 3D-2D correspondences with the Direct Linear Transform [2]. Similarly, Scaramuzza *et al.* [200] uses a 3D laser sensor to obtain highly accurate camera intrinsics. In contrast to this, our application has to work with a single pose video, where the factory-calibrated intrinsics are further known to be inaccurate. Furthermore, dynamic scenes and texture-less regions lead to poor point correspondences. To alleviate these challenges, our approach utilizes the technique proposed by Alvarez *et al.* [8], which minimizes an energy objective based on rectifying straight lines that are present on the soccerfield.

PARTICLE SWARM OPTIMIZATION A core component of our method is the fusion of different sources of information through optimization. In such scenarios, the best objective weights of the optimization problem are often obtained using an exhaustive grid search. However, this process is computationally expensive and requires discretizing the search space. As an alternative, meta-heuristic algorithms such as simulated annealing [124] and particle swarm optimization (PSO) [184] have shown good results in various domains [98, 106]. In our approach, the PSO algorithm is used for the constrained optimization of the weights for different cost terms.

5.2 METHOD

In this section, we detail our pipeline for consistent player tracking and identification. Figure 5.1 provides an overview of the key components in our target application. Our pipeline consists of three parts. First, the camera intrinsics and extrinsics are estimated using field features, such as lines and corners, whose dimensions and relative positions are known a priori. Then, player tracklets are generated and the jersey color is estimated for each tracklet. Additional information, such as the players’ self-estimated position and game state are extracted from the game logs. The final step associates each tracklet with a specific robot player. We perform this crucial step by optimizing a binary quadratic program. The performance is further improved by finding the best cost weighting using PSO. In the following, each component is described in detail.

5.2.1 *Data and Application*

We consider RoboCup Soccer SPL matches between teams of 5 NAO robots, where data is acquired from an external camera as well as the game-log. The game-log is generated by the Game Controller, which communicates the game state (start, end, free-kick, player penalties) to the players through WiFi. Furthermore, each player is required to send a heartbeat network packet including its estimated position to the Game Controller at 1Hz. These are logged by the Game Controller together with the game states. In addition, players can exchange information with their team members by broadcasting network packets at a fixed rate. These are also captured and logged by the Game Controller. Our dataset is composed of 8 annotated 5000-frame sequences recorded with a wide-angle camera at 30 FPS and the Game Controller logs of the corresponding matches. The sequences were extracted from videos recorded at RoboCup 2019 and 2022, and the frame timestamps have been synchronized with the Game Controller logs. The annotations include the bounding box, jersey color and number of each active player which is visible on the field in each frame. The object detection and image classification models and the optimizer’s weights are trained on five of these sequences. The remaining three sequences are used for evaluation.

5.2.2 *Camera Calibration and Pose Estimation*

Accurate camera calibration and pose estimation is essential in our method to locate robots in the image on the field and to remove false positive detections outside the field boundaries. Estimating the radial distortion coefficients is especially important in this case due to the barrel distortion introduced by the wide angle lens.

We assume a static camera over the sequence, which is the setup used for all RoboCup game recordings. We, therefore, use the known geometry and dimensions of the field lines for the camera calibration. The main pre-requisite for the task is to establish clean images with clear correspondences between the target frame and the known 3D geometry. Due to moving robots and humans on the field, occlusions are present. We resolve these and obtain a clean unoccluded view of the field by computing the median image over the whole sequence.

Widely used calibration algorithms that are implemented in common computer vision toolboxes require either multiple views of a flat calibration

target [256] or several accurate 2D-3D correspondences of non-coplanar points on the calibration target. In our application, however, the former approach is not applicable due to the lack of camera motion. We further observe that the later algorithms, based on 2D-3D correspondence fail to jointly estimate distortion coefficients, intrinsics and extrinsics. This is due to the low number of available calibration points and missing good initial estimate of the distortion coefficients. Therefore, we approach the problem in two steps:

First, we estimate radial distortion coefficients by leveraging the fact that field lines should be straight. Groups of points belonging to the same field lines are selected and used to formulate the optimization problem according to Alvarez *et al.* [8].

In the second step, extrinsics are computed and the focal lengths are refined if needed. To this end, we leverage the known 3D soccer field landmarks, specifically the line intersection positions. After undistorting the median image using the parameters estimated in the previous step, line segments are detected with the SOLD2 [178] line detector. To filter out initial false positives, a mask of the field area is estimated using color thresholding. Lines are then further refined it with morphological dilation followed by the Spaghetti algorithm [25] for connected component labeling. The remaining line segments are merged into large and straight field lines by clustering them based on their proximity of endpoints and collinearity [217].

Intersections are computed from the detected and post-processed lines, which provides the required 3D-2D point correspondences to the ground truth 3D field coordinates. Altogether, we obtain 7 reliable point pairs in each of the videos. In order to compute the camera poses, the P3P [126] algorithm followed by a non-linear refinement step is utilized. Although the non-linear refinement can potentially further improve the intrinsics, we find that the intrinsics are already accurate enough for our purpose at this stage.

5.2.3 *Multi Object Tracker*

To generate bounding box tracklets, we use Tracktor [18] with Faster-RCNN [190] with Feature Pyramid Networks (FPN) and a ResNet-50 backbone. We initialize the model with MS-COCO [144] pre-trained weights and fine-tune it on the training sequences of our dataset to detect robot players. Since during matches players often occlude each other for several seconds, we set the patience of the tracker to 1 and use conservative thresholds for

the NMS step to prevent tracklets from switching from one player to another. In this way, when players cluster in one area of the field and occlude each other, several short-lived tracklets are initialized. Our optimizer is then able to robustly combine these into longer tracks.

Subsequently, the trajectory of each player tracklet is converted to field coordinates, (x, y) . We approximate the position of the robots' feet by the midpoint of the lower side of each bounding box. This point is then projected to field coordinates using the camera pose estimated during calibration to obtain 2D positions in field coordinates. Each resulting projected tracklet j is smoothed using a Kalman filter with a constant velocity model.

5.2.4 Jersey Color Detection

In the SPL, 9 distinct jersey colors are used. These colors, known for each match, provide a strong signal to associate tracklets with players from either team. We thus, train a VGG16 network to detect jersey colors for each tracklet and assign a score for each of the team colors. As the colors of the two playing teams are known, only predictions for these are considered at this stage.

5.2.5 Robot States

The Game Controller logs include several sources of information about the state of the active players at every point in the game. In our formulation, we make use of information from the following states to match tracklets to players:

Self Localization: The robots calculate their position on the field based on the field landmarks they observe with the onboard cameras. The estimated positions are often sufficiently accurate and can be correlated with tracklet trajectories to provide a strong signal for identification. However, relying on this signal alone is not possible, as they can diverge arbitrarily far from the true value due to drastic changes in lighting conditions or other factors like the players losing track when falling over.

Fallen Robot: The robots use the IMU information and heuristics to determine when they fall. In the external camera, when a player falls, its bounding box has an aspect ratio higher than 1. Therefore, a player tracklet whose bounding box has an aspect ratio higher than a given threshold for a certain number of consecutive frames is considered to be a fallen player

tracklet, which can be matched to the robots' internal states and provides another strong signal.

Penalties: In RoboCup soccer, the robots are penalized and removed from the field if they fail to follow the game rules, e.g. if they commit a foul or suddenly start leaving the field. These events are used to add constraints to the problem to prevent the optimizer from matching an active tracklet to a penalized player.

5.2.6 Global optimization

Even though there are at most 10 active robots in the considered soccer matches, occlusions and distractors cause Tracktor to split the tracks into a large number of tracklets. Therefore, we frame the long-term tracking problem as an assignment of *tracklets* to a fixed number of player *tracks*. It is modeled as a constrained quadratic binary optimization problem. We denote the index set of player tracks $I = \{1, \dots, N\}$ (with $N = 10$) and generated tracklets $J = \{1, \dots, M\}$. The objective is to minimize:

$$H(x) = \sum_{i \in I} \sum_{j \in J} x_{i,j} (O_u + \sum_{l \in L} w^l c_{i,j}^l) + \sum_{i \in I} \sum_{j \in J} \sum_{k \in J} x_{i,j} x_{i,k} (\sum_{p \in P} w^p c_{j,k}^p), \quad (5.1)$$

where $x_{i,j} \in \{0, 1\}$ are binary optimization variables, with $x_{i,j} = 1$ meaning tracklet j is assigned to track i , L and P is the number of unary and pairwise cost functions, $c_{i,j}^l$ the unary (tracklet-to-track) costs, and $c_{i,j,k}^p$ the pairwise (tracklet-pair-to-track) costs. The scalars w^l, w^p are the *cost weights* and the scalars O_u denote the offsets. The offsets are negative to penalize the trivial solution of assigning nothing ($x_{i,j} = 0 \forall i, j$).

To prevent generating invalid tracking solutions, the following constraints are implemented. The first set of constraints

$$\sum_{i \in I} x_{i,j} \leq 1, \forall j \in J, \quad (5.2)$$

prevents assigning a single tracklet to multiple tracks. The second set of constraints is implemented to avoid merging temporally overlapping tracklets

$$x_{i,j} x_{i,k} = 0 \forall i \in I, \forall (j, k) \in J \times J : T_j \cap T_k \neq \emptyset, \quad (5.3)$$

where T_j, T_k represent the set of frames in which detections exist for tracklets j and k respectively.

5.2.7 Cost terms

Our formulation uses two types of cost terms: 1) Unary cost terms are a measures for the fit between tracklets and tracks. 2) Pairwise cost terms that measure the fit between pairs of tracklets. Overall, we utilize the following cost-terms:

Self-localization - During the matches each robot sends its estimated position on the field (x, y) once per second. The signal is linearly interpolated between timestamps and the distance to the position estimated from the external camera is computed. Averaged over each tracklet, this provides a strong prior for the assignment problem. To encourage matching a tracklet to a player's track when its trajectory is close to the player's communicated trajectory, we define the following cost term:

$$c_{i,j}^{loc} = \frac{\beta^{loc}}{|T_j|} \sum_{t \in T_j} \|\hat{\tau}_j^t - \tilde{\tau}_i(t)\| \quad (5.4)$$

where β^{loc} is a scaling factor.

Jersey color detection - Let \bar{p}_j^H and \bar{p}_j^A be the mean probabilities of tracklet j belonging to a player of the **Home** or **Away** teams respectively. We encourage matching tracklets to the correct team with:

$$c_{i,j}^{team} = \begin{cases} 1 - \bar{p}_j^H & \text{if } i \in I_H \\ 1 - \bar{p}_j^A & \text{if } i \in I_A \end{cases} \quad (5.5)$$

Fallen robot state - Fallen player tracklets detected with the heuristic described above can be easily matched to fallen player events in the Game Controller logs. Given a fallen robot event reported by player i recorded in a given time frame, for each fallen robot tracklets detected in the same time frame we add a fixed cost term $c^{fallen} = 1$ to discourage matching these tracklets to other players.

Duration - To filter out false positive tracklets, which are usually short, we use the following cost term to encourage matching with longer tracklets:

$$c_{i,j}^{duration} = \min(1, \frac{\mu}{T_j}) \quad (5.6)$$

where μ is a tunable threshold.

Global trajectory continuity - A pair of consecutive non-overlapping tracklets (j, k) is more likely to belong to the same track if the earlier tracklet "ends near" the start of the later tracklet. We extrapolate the pose of the earlier tracklet j from its end position using a constant velocity model:

$$\hat{\tau}_j(t) = (\hat{x}_i^{t_{j,f}}, \hat{y}_i^{t_{j,f}})^\top + (t - t_{j,f})(\hat{v}_i^{t_{j,f}}, \hat{w}_i^{t_{j,f}})^\top \quad (5.7)$$

We define the following pairwise cost term based on the distance to the start of any temporally close tracklet:

$$\begin{aligned} c_{i,j,k}^{cont} &= \|\hat{\tau}_j(t_{k,i}) - \hat{\tau}_k^{t_{k,i}}\| \\ \forall i \in I, (j, k) \in J \times J : 0 < t_{k,i} - t_{j,f} < \theta_{cont} \end{aligned} \quad (5.8)$$

where $t_{j,f} = \max(T_j)$, $t_{k,i} = \min(T_k)$, $\hat{\tau}_k^{t_{k,i}}$ is the earliest pose of tracklet k , and θ_{cont} is a tunable parameter.

5.2.8 Optimization of Cost Weighting

The weights for each cost term and the offsets define the optimization problem and thus the performance of the tracking results. Assigning a high weight to a cost term ensures that the optimizer pays more attention to that attribute, while the offsets implement an error threshold for tracklet assignment. We use PSO for optimization the weights for the cost terms using the ground truth training data to maximize the metrics. and initialize particles randomly over the search space. The weights corresponding to the different cost terms and the offset are the values represented by each particle. At each step, the values of the particles that correspond to the cost terms are updated such that they are non-negative and their sum is normalized to 1. This ensures that no redundant information is modeled. A the same time, the value corresponding to the offset for each particle is kept negative. For all experiments, 50 particles are initialized randomly and the optimization is run for 100 iterations. The cognitive parameter and the social parameter, which control a particle's affinity to its best position and the global best position are kept at 2 for the whole run. The objective function is the average of the Mean Player Identification Recall (MPIR) for all the sequences and the optimization aims to maximize it. The search stops when the number of iterations are completed or when all particles have converged to the same position.

5.2.9 Reference Method: DeepSORT

While our task provides information beyond what common tracking pipelines are able to utilize, it is important to quantify the performance relative to existing trackers. To fulfill the task of long-term tracking and player identification we augment the DeepSORT tracker [233, 234] by a greedy tracklet matching algorithm.

DeepSORT is an extension of the optimization-based SORT algorithm [20] that integrates appearance information from a pre-trained network to create a deep association metric. This metric combines motion and appearance cues to establish measurement-to-track associations during tracking. Motion cues are integrated through Kalman filtering and data association is performed using the Hungarian algorithm.

While DeepSORT can handle occlusions by using the re-identification module, it commonly is only able to do so over short timeframes. We, therefore, combine it with a *greedy tracklet matching* approach. Any tracklet which has not been assigned is matched with the spatially closest inactive track. Furthermore, constraints are applied to prevent multiple tracklets being assigned to the same track if they have any time overlap. At the start of each run, the total number of robots which are present in that session is provided for initialization. This provides additional privileged information and can bound the maximum number of tracks which are generated.

Furthermore, a pure tracking pipeline like DeepSORT is able to generate long-term tracks, but cannot detect the ID of each robot. To circumvent this issue, we manually assign the first tracklet appearing for each robot to the corresponding ground-truth ID, which forms an oracle approach for identification. I.e. a perfect tracker would also perform perfect identification using this approach. While this provides additional information beyond what is used in our method, it allows us to compare our method to a fair tracking-baseline that uses the best-possible identification approach.

Finally, to allow for a well performing baseline and fair comparison, we tune the DeepSort baseline re-identification time over the testset. Table 5.3 shows the different metrics for different values of re-identification time.

The algorithm's performance shows an increase with the increase in the number of frames within the reidentification window. However, when it is too high, the performance degrades, as the initial tracklets are more likely to contain ID-switches and therefore contain errors that cannot be corrected later. Based on this, we select a maximum re-identification time of 900 frames corresponding to 30s of video for our baseline method.

Full	Self Loc.	Tracklet Duration	Team Det.	Fallen Flag	Penalized Flag	Tracklet Distance
88.11	15.39	51.14	76.48	86.22	76.27	83.33

TABLE 5.1: The ablation study evaluates the influence of removing different information used to match tracklets to tracks. All numbers are provided in percent MPIR on the testset.

Method	MPIR	Time Frame	30	150	300	900	1800	3600	5400
		ours	88.11						
Oracle Deepsort [234]	43.76	MPIR	42.45	42.31	40.08	43.75	38.40	38.51	38.51

TABLE 5.2: Results on the test-set for our approach and the extended deepsort baseline. All values are provided in percent.

TABLE 5.3: DeepSort Performance with different re-identification time.

5.3 EXPERIMENTAL RESULTS

We evaluate our approach over a test set of 3 sequences of 5000 frames recorded at 30 frames per second. Each video covers a different game, thus testing our approach with different levels of player self-localization accuracy, team colors, and environmental conditions. Since we are primarily interested in correctly identifying players in every frame, commonly used MOT metrics such as MOTA are not relevant, as they do not measure player identification performance. Therefore, we define an ad-hoc metric more suited to our problem setting, the MPIR. For each frame t in a sequence, we match the bounding boxes predicted by the tracker to the ground truth based on an IoU-threshold of 0.5. We denote with TP_t the number of correctly identified bounding boxes and with FN_t the number of incorrectly identified bounding boxes. The tracking metrics then read as:

$$\text{MPIR} = \frac{1}{T} \sum_{t=1}^T \frac{TP_t}{TP_t + FN_t}. \quad (5.9)$$

Table 5.1 shows the MPIR, the ratio of times each player has been identified correctly. The first column shows our full approach. Subsequent

columns show ablations, with each feature removed separately. The cost weightings are optimized using PSO for each scenario. Figures 5.2 and 5.3 provide a visual representation of the results obtained with our algorithm on a sequence from a match played at RoboCup 2019.



FIGURE 5.2: Visualization of robots identified by the tracker. The tracking result is represented by bounding boxes and IDs at their top. Ground truth positions are represented by green crosses and corresponding green IDs.

5.3.1 Ablation Study

We perform an ablation study and depict results in Table 5.1. With all features, we achieve 88.11% MPIR. Removing the robot self localization has the strongest impact with 15.39% MPIR remaining, while removing the fallen robot flag results in the least performance drop. This is expected since the self-localization is an important attribute that provides information about the position of the robot in the field and consequently the image. The fallen robot flag is noisy, as it relies on the robot’s IMU and an approximate heuristic to detect whether the robot has fallen in the video.

5.3.2 Feature Importance

We can further analyze each feature’s importance through the weights obtained from the PSO optimizer, where a higher weight indicates higher importance. Figure 5.4 shows the importance of the features in each column for the different ablations represented by each row of the table. The first row corresponds to the full model and each following row to one of the ablations where a single feature weight is set to zero.

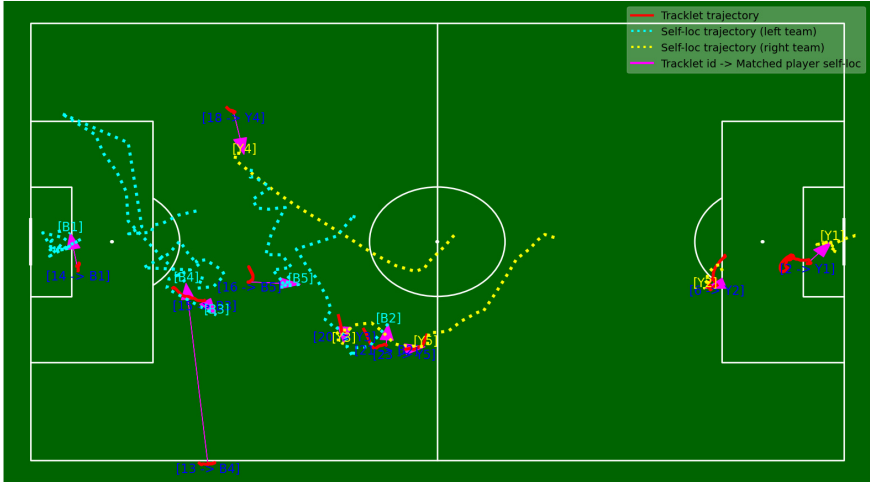


FIGURE 5.3: Top-view of Figure 5.2. The dotted lines represent the players’ self-localization trajectories. The red lines are the tracklet trajectories in field coordinates. For each tracklet, the original ID and the matched player ID are shown. The purple arrows connect each tracklet trajectory to the self-localization trajectory of the player to which the tracklet has been matched.

While the weighting of different features needs to be handled with care due to their scaling, we can compare the weights of the same feature in different ablations directly. Strong weights are assigned to the self-localization and tracklet duration, which provide strong indicators for matching and tracklet confidence. Removing these features shows that weighting is redistributed: While in the full model the noisy fallen robot events are not used, they are incorporated when no self-localization information is available. In this case, the primary source of information to match tracklets to robot IDs is missing but can be replaced by matching the fallen robots.

5.3.3 Explainability

Using several sequences for the optimization of cost weights yields weights which can generalize to new data. However, since the matches are played by different teams which have non-identical algorithms running on the robots, the weights might be suboptimal for some matches. By searching for the parameters which yield the best results on a single sequence, we can further understand the shortcomings and types of noise exhibited by each team.

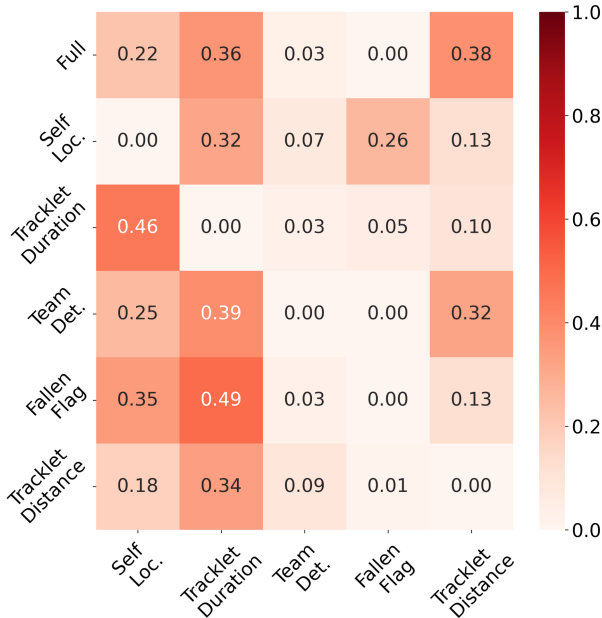


FIGURE 5.4: Feature weights for ablated features.

This further allows us to better understand and explain the inner workings of the proposed algorithm.

For this purpose, we use PSO to optimize the weights individually for each sequence in the test set to find the optimal cost weights. Then we compare the resulting parameters, reported in Table 5.4, with qualitative observations on the game-videos itself. We discuss the outcome of this analysis in the following.

Sequence 6 - In this sequence, the self-localization cost is given a relatively high weight. This sequence was extracted from the 2022 championship final between the top teams in the league and it was played under ideal lighting conditions, so the players' self-localization is accurate. Thus, a higher weight on this feature is expected. However, throughout this sequence, several players are penalized and manually moved outside of the field, which causes their internal position estimate to diverge and match less closely the tracklets trajectories. The jersey color detection is also assigned a comparatively high weight. The jersey colors of the two teams are different from each other and can be detected very accurately.

Seq #	Self-Local.	Jersey	Track Time	Fallen	Traj. cont.	Offset
6	0.27	0.16	0.34	0.02	0.22	-0.19
7	0.20	0.0	0.4	0.04	0.36	-0.45
8	0.35	0.28	0.23	0.0	0.14	-0.2

TABLE 5.4: Cost weights optimized individually for each sequence in the test set with PSO.

Finally, we can observe that tracklet duration is also given high importance. In this sequence, the players are completely occluded by the referees at several points in the sequence, during which many short-lived false-positive tracklets are created. However, since the players are distributed evenly on the field the rest of the time, there are also several long-lived tracklets. Since matching these correctly can significantly affect the metrics, prioritizing this feature helps the identification process.

Sequence 7 - In this sequence, the self-localization weight bears the lowest weight of all three sequences. This is because the self-localization is accurate for one of the teams, but is often very noisy and incorrect for the other. The players are rarely occluded by the referees and the players do not often cluster in one part of the field as often seen these matches. As a result, there are several long lived and very accurate tracklets, hence the higher importance given to tracklet duration. The large offset term is most likely related to the low occurrence of tracklet switching and detection false positives, the optimization. Hence, this term discourages the optimizer from discarding tracklets.

Sequence 8 - In this sequence, the self-localization has a high weight. Self-localization is accurate for one team (black jersey), but is rather unreliable for the other team (yellow jersey) because of a software malfunction causing the players to often report their position to be in the center of field. However, several players of the later team are penalized for the game for most of the sequence, which means no tracklets are assigned to them due to the constraints. As a result, most of the tracklets represent the black team, resulting in a good self-localization performance, which makes it an important feature. Jersey color detection also carry high importance in this sequence. This is because the jersey colors of the two teams are strongly distinct and easy to detect, such that team detection can easily help differentiate tracklets belonging to players of different teams.

5.4 CONCLUSION

In this chapter, we presented a sensor fusion-based method for tracking multiple similar humanoid robots. We utilize information from both visual data and their own sensors by combining tracklets using a quadratic optimization technique. The method allows automated tracking of robots over a long time on a stationary video sequence. Open points that we will investigate in the future include the evaluation in more complex environments as well as the interpolation of tracks during occlusions.

Part III

QUANTUM COMPUTER VISION AND
MACHINE LEARNING

Over the last years, the computer vision community has approached a large number of real-world challenges and pushed these problems from the domain of fundamental research to a state that allows to build products on top of it. This comes at the cost of requiring powerful computational resources during the training of networks, for inference and for solving optimization problems. Quantum computing provides a long-term perspective to solve this hunger for compute, as it can efficiently solve a range of complex and hard problems.

One kind of task that reappears throughout many computer vision applications is discrete optimization, which often is NP-hard thus, cannot be solved exactly on large problem instances. In Chapter 6 we derive a quantum computing formulation of MOT, where a quantum computing-based optimizer is used to solve the track assignment problem. In Chapter 7 we further investigate the efficient use of the quantum computer by using all solutions measured on the system to quantify the confidence of clustering solutions.

III.1 RELATED WORK

With the availability of quantum computers to the general research community [29, 52, 159, 160], the research interest in finding applications for such systems has considerably increased. In this context, adiabatic quantum computing (AQC) [29, 111] provides a well-tangible starting point, even though many applications need a complete reformulation considering the architectural differences of a quantum computer. Current applications for AQC include solving optimization problems for improved traffic flow [172], robotic routing problems [175], the design of molecules [170] or portfolio optimization in the finance sector [168]. In other fields like physics, quantum computing has been used to reconstruct particle interactions [49] or to engineer genes [73].

Recently, the computer vision community has developed a strong interest in finding applications for quantum computing, which are related to hard permutation problems [16, 17, 23] that are solvable using AQC or Quantum Annealing (QA). Non-maximum suppression (NMS) for object detection has been formulated as a quadratic unconstrained binary optimization (QUBO) problem solvable using quantum computing [138] and in 3D vision quantum computing has been successful for a range of shape and point-matching tasks [16, 21, 163] and for optimizing geometry compression [67]. All of these approaches are closely related to optimization for computer

vision [129, 246]. In this context, one of the tasks of special interest for the community is model fitting which finds applications in estimating camera parameters [59] or separating different motion components [10]. It is traditionally solved using consensus maximization but can be reformulated as a quantum-computing problem [10, 40, 41, 59, 66, 235].

While AQC implements a universal quantum computer in theory, current hardware implementations do not contain all required couplings between qubits and thus, are still limited in their capability. In contrast to this, circuit-based quantum computing implements universal quantum computers in hardware already today [7, 30, 103, 236]. Using this computing paradigm, the 2D signal processing community has proposed a range of encoding schemes that efficiently represent images using qubits. FRQI [132] uses an amplitude encoding for color images, where the probability amplitude of each state represents one greyscale value. This requires $\lceil \log N \rceil$ qubits to encode N pixel locations and one qubit to store the brightness, corresponding to the minimum number of qubits [145], without applying further compression. This comes at the cost of deep circuits being required to prepare the state of the qubits, which is a strong limitation with current quantum computers. NEQR [255] achieves a quadratic speedup on the quantum image preparation by representing the quantized grayscale value of pixels as the basis-state of a qubit sequence and not as probability amplitudes of a single qubit. Further approaches and modifications [109, 136, 137, 198, 209, 223] aim at improving the space efficiency of encoded images as well as at reducing the depth of circuits needed to generate encodings.

Le *et al.* [133] propose approaches to perform a set of geometric transforms using the FRQI [132] representation and NEQR [255] further discusses basic operations that can be performed on the quantum image representations. For many fundamental machine-learning tasks, approaches using a quantum computer have been proposed [22, 112], that have the potential to provide a speedup or better representative power with a lower number of qubits. For deep learning on image data, Pan *et al.* [177] propose a neural network layer based on the Hadamard transform that allows to implement 2D convolutional layers efficiently on a quantum computer. Clustering is another well-studied machine-learning problem for quantum- as well as quantum-inspired algorithms [5]. Quantum clustering [94] uses the Schrödinger equation to model the clustering problem, where cluster centers are defined as the minima of the corresponding potential function. Casaña-Eslava *et al.* [37] extend this formulation with a probabilistic estimate of cluster memberships. In [231] K-NN is implemented on a quantum

computer based on encoding each point and computing Euclidian distances on the system. Bermejo and Orus [19] state clustering as an optimization problem that is tailored towards a variational quantum eigensolver. These approaches use classical computation and gate-based quantum computers that are currently still on a small scale.

Closest to the work presented in this thesis, Arthur *et al.* [11] propose a balanced k-means clustering algorithm suitable for an AQC and Nguyen *et al.* [174] use a clustering approach to group visual image features. Our underlying clustering algorithm follows the same approach as [11]. However, while they discard all but the best measurement, our approach utilizes all information by employing AQC as a sampler to generate probabilistic solutions of the clustering problem, rather than only using the best solution in an optimization framework.

III.II BASICS OF QUANTUM COMPUTING

Quantum computing is a fundamentally new approach, that utilizes the state of a quantum system to perform computations. In contrast to a classical computer, the state is probabilistic and described by its wave function, which enables the use of fundamental properties of quantum systems like superposition and entanglement. By exploiting these properties, a range of problems that quickly grow in complexity on classical computers and thus, cannot be solved in any reasonable timeframe, could be solved considerably faster [68] by a quantum computer. Reaching such a point is widely referred to as quantum primacy. Even though implementations of quantum computers are still heavily experimental, some problems have already been shown to profit from them, including the sampling of pseudo-random quantum circuits [12, 237] and Gaussian boson sampling [259]. While these tasks are of a strong academic nature, several algorithms approach important and impactful applications. The most well-known examples in this domain are the prime factorization algorithm by Shor [205] and Grover's database search algorithm [81]. To provide a fundamental overview, the following section introduces the basics of quantum computing.

Qubits are two-state quantum-mechanical systems that form the basis of quantum computers. Like a bit, a qubit has two basis states that can e.g. be $|0\rangle = [1\ 0]^T$ and $|1\rangle = [0\ 1]^T$, which in a superposition form the qubit's state. Qubits can be implemented using different quantum-physical systems, depending on the required use-case. Examples include superconducting circuits, as in the quantum computer used in this work, ions trapped in

an electromagnetic field, or photons where the polarization represents the qubit state.

Quantum Superposition refers to the property of a quantum system that it is not required to be in one of the basis states, but rather can be described by a linear combination of possible basis states. A qubit in a pure state $|\psi\rangle$ can be described with its two basis states $\{|0\rangle, |1\rangle\}$ as

$$|\psi\rangle = c_1 |0\rangle + c_2 |1\rangle \quad (5.10)$$

where c_1 and c_2 are complex numbers, called probability amplitudes, with $|c_1|^2 + |c_2|^2 = 1$.

Measurement. During computation on the quantum computer, the state of the system can be any valid superposition of basis states. However, a measurement of the system always results in a single basis state. The probability of measuring a state is the respective squared amplitude. In the single qubit case, this corresponds to

$$p(|0\rangle) = |\alpha|^2 \quad p(|1\rangle) = |\beta|^2. \quad (5.11)$$

As a measurement corresponds to an observation of the qubit it leads to wave function collapse [232], which means that the qubit state is changed irreversibly [80]. Which is in contrast to all other operations in quantum computing.

Entanglement of qubits is at the very heart of quantum computing [110]. A system of entangled qubits is represented by a system state where each qubit cannot be described only with its own state but depends on the state of the remaining system [63, 89, 127, 151, 203]. This implies that it is not possible to decompose the joint state into the tensor product of each separate qubit state. Therefore, measuring the state of one qubit in an entangled system affects the state of the other qubits [80]. This is a fundamental property of quantum mechanics and plays a crucial role in the exponential scaling capabilities of quantum computing.

Adiabatic Quantum Computing. AQC, which is used to solve our formulation, is a quantum computing paradigm where the state of a quantum system is modified by performing an adiabatic transition. Current hardware implementations such as the D-wave systems [29] follow this approach and implement a QA to solve QUBOs. They are based on the Ising model [104, 111], which describes the configuration of a set of interacting particles that all carry an atomic spin σ_i .

The spin can either be $+1$ or -1 and the particles are coupled by interactions J_{ij} as well as influenced individually by a transversal magnetic field h_i . The energy of this system is described by its Hamiltonian function

$$H(\sigma) = - \sum_i \sum_j J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i. \quad (5.12)$$

Thus, finding the lowest energy state or ground state of the Ising model corresponds to solving the QUBO defined by the Hamiltonian function. This relation is used in the QA, where the system of qubits implements an Ising model H_T that represents the QUBO of interest. Such problems are often NP-hard and known to be very challenging for classical solvers. As this task can directly be implemented on an adiabatic quantum computer, a considerable speedup for large problem instances is expected in the future.

The lowest energy state is found by following the adiabatic theorem [26]. Starting with an initial system in its ground state described by a Hamiltonian H_0 , the adiabatic theorem states that during a sufficiently slow change of the Hamiltonian, the system never leaves its ground state. The change of the Hamiltonian

$$H(t) = H_0 \left(1 - \frac{t}{T_a}\right) + H_T \frac{t}{T_a} \quad (5.13)$$

is called an adiabatic transition. The transition is performed over the annealing-time T_a and allows to solve an optimization problem with the Hamiltonian H_T . The condition for a sufficiently slow evolution depends mostly on two factors, the temperature of the environment and the spectral gap of the Hamiltonian, i.e. the difference between lowest and second-lowest energy level or eigenvalue. While the first is a system property, the second can be influenced by choosing a suitable Hamiltonian [17].

While in an ideal noise-free case, the system stays in its ground state, any real system is embedded in a temperature bath that can induce a change to a higher energy state. The distribution of measured final states will then follow the Boltzmann distribution with temperature T

$$p(\sigma) = \frac{\exp[-H(\sigma)/T]}{\sum_{\sigma'} \exp[-H(\sigma')/T]}, \quad (5.14)$$

where $p(\sigma)$ describes the probability of finding the system in state σ and σ' are all possible states.

ADIABATIC QUANTUM COMPUTING FOR MULTI-OBJECT TRACKING

MOT is most often approached in the tracking-by-detection paradigm, where object detections are associated through time. The association step naturally leads to discrete optimization problems. As these optimization problems are often NP-hard, they can only be solved exactly for small instances on current hardware. AQC offers a solution for this, as it has the potential to provide a considerable speedup on a range of NP-hard optimization problems in the near future. However, current MOT formulations are unsuitable for quantum computing due to their scaling properties.

We therefore propose the first MOT formulation designed to be solved with AQC. We employ an Ising model that represents the quantum mechanical system implemented on the AQC. We show that our approach is competitive compared with state-of-the-art optimization-based approaches, even when using off-the-shelf integer programming solvers. Finally, we demonstrate that our MOT problem is already solvable on the current generation of real quantum computers for small examples, and analyze the properties of the measured solutions.

6.1 INTRODUCTION

MOT is a task in computer vision that requires solving NP-hard assignment problems [95, 96, 214]. To make this feasible, the community proposed a range of different approaches: work on the problem formulation using domain knowledge helps to make it easier to solve problem [95, 214], approximate solvers extend the feasible problem size [96], and the combination of deep learning with simple heuristics can be seen as a data-driven approach to the problem [28, 48]. Nevertheless, integer assignment problems remain hard optimization tasks for any available solver. With the recent progress in quantum computing, a new way of solving such optimization problems becomes feasible in the near future [9, 122, 219].

Instead of iteratively exploring possible solutions, e.g. via branch and bound, the problem is mapped to a quantum mechanical system, whose energy is equivalent to the cost of the optimization problem. Therefore, if

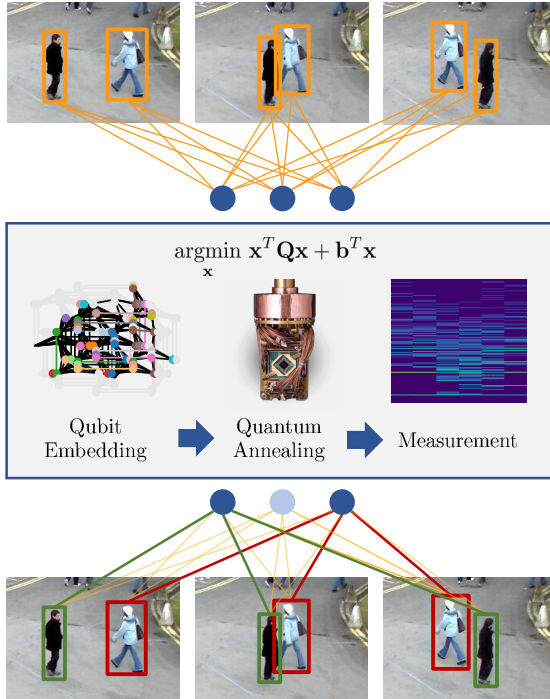


FIGURE 6.1: The proposed approach to MOT states the assignment problem between detections and a set of tracks as a quadratic unconstrained binary optimization task. We then represent the optimization problem as a quantum mechanical system that can be implemented on an AQC. Via quantum annealing, a minimum energy state is found that represent the best assignment.

it is possible to measure the lowest energy state of the system, a solution to the corresponding optimization problem is found. This is done with an AQC, which implements a quantum mechanical system made from qubits and can be described by the Ising model [104]. Using this approach, a quantum speedup, which further scales with system size and temperature, has already been shown for applications in physics [121, 122].

While there is a range of advantages that quantum computing can provide in the future, mapping a problem to an AQC is not trivial and often requires reformulating the problem from scratch, even for well-investigated tasks [16, 23]. On the one hand, the problem needs to be matched to the Ising model, on the other hand, real quantum computers have a very limited number of

qubits and are still prone to noise, which requires tuning of the model to handle the limitations.

In this chapter, we present the first quantum computing approach to MOT. The number of required qubits in our formulation grows linearly in each the number of detections, tracks¹ and timesteps and only requires additional couplings between qubits to model long-term relations. Our overall contributions are the following:

- A quantum computing formulation of MOT that is competitive with state-of-the-art methods.
- A method using few problem measurements to find Lagrange multipliers that considerably improve solution probability.
- Extensive MOT experiments on synthetic as well as real data using a D-wave AQC.

The remaining chapter is structured as follows: After presenting related work, the basics of quantum computing are introduced. This is followed by our MOT formulation that is optimized to run on an AQC. We then show the changes required to make the problem solvable also with the classical computing paradigm. Finally, experiments on a D-wave quantum computer are presented together with results on larger problem instances.

6.2 QUANTUM MOT

Most existing optimization-based approaches to MOT aim at finding feasible relaxations [96], implement efficient heuristics in the solution approach [95] or use deep learning together with post-processing [28] to solve the assignment problem. With the considerable amount of work invested into them, the problem became solvable for growing instances by now. Nevertheless, the assignment problem stays an NP-hard task to solve and growth is thus limited. Quantum computing with the associated speedup on hard problems can provide a solution to this challenge, even if the corresponding optimization problem is much harder to solve with classical approaches at the moment. However, representing tasks in a form suitable for quantum computing often requires a completely new formulation of the problem and MOT is not different in this aspect.

While widely used flow formulations [95, 96, 140] are suitable for exploiting sparsity, they come with a large set of inequality constraints, which

¹ It is important to note that with additional detections also a larger number of tracks might be required, which further increases the number of required qubits.

makes them intractable on near-future quantum computers that are limited in the number of qubits. In this context, permutation matrices were shown to be a powerful tool for synchronization or shape matching [16, 17, 23].

MOT FORMULATION. We approach the MOT problem following the tracking by detection paradigm and use a fixed set of available tracks. Given a set of detections in each frame of a video, appearance features are extracted for each detection. By using a multi-layer Perceptron, pairwise appearance similarities between detections at different timesteps are computed [96]. Starting with this, the goal of the tracking algorithm is to assign each detection to a track, such that the sum of the similarities of detections assigned to a single track is maximized. In this context, a track is defined by its track ID t and each detection in a frame f by its detection ID d .

We formulate the given task of assigning detections to a joint set of tracks using assignment matrices, which relax the assumptions of permutation matrices. The binary assignment matrix \mathbf{X}_f for a frame f maps a vector of detection indices to a vector of tracks at every frame of a video. The elements $x_{dt} \in \{0, 1\}$ of the assignment matrix represent the connections between detections d and tracks t . Given $D - 1$ detections and $T - 1$ tracks, the assignment matrix assigns a detection to a track if $x_{dt} = 1$. The requirement that a single detection is assigned to a track at one timestep, leads to the constraint

$$\sum_{d=1}^D x_{dt} = 1 \quad \forall t \in \{1, \dots, T - 1\}. \quad (6.1)$$

And reversely, Equation 6.2 asserts that every detection is assigned to a single track

$$\sum_{t=1}^T x_{dt} = 1 \quad \forall d \in \{1, \dots, D - 1\}. \quad (6.2)$$

To allow for false-positive detections as well as to handle the case of fewer detections than available tracks, one dummy-detection and one dummy-track, with the respective indices D and T , are introduced. A detection assigned to the dummy-track is treated as a false positive and a track that got the dummy-detection assigned to it is inactive or occluded. As the dummy-track and dummy-detection may be assigned multiple times, constraints 6.1 and 6.2 do not apply to them. To model tracks in a sequence consisting of F frames, a single assignment matrix \mathbf{X}_f is required for each frame f , mapping the detections to tracks.

QUADRATIC FORM. The basis for optimization-based trackers are costs between pairs of detections, where the cost is accounted for if two detections are connected by a common track. The goal of the tracker is to find a solution that minimizes the total cost associated with the assignment. Our approach using assignment matrices leads to a quadratic cost for a pair of frames i, j that reads

$$c_{ij} = \sum_t \sum_{d_i} \sum_{d_j} x_{id_it} q_{d_id_j} x_{jd_jt}, \quad (6.3)$$

with x_{id_it} and x_{jd_jt} being entries from the assignment matrices \mathbf{X}_i and \mathbf{X}_j respectively and $q_{d_id_j}$ as the corresponding similarity score. It is important to note that only detection pairs assigned to the same track incur a cost, which results in a single sum over the tracks t .

Equation 6.3 can be written in matrix form as

$$c_{ij} = \text{vec}(\mathbf{X}_i)^T \mathbf{Q}_{ij} \text{vec}(\mathbf{X}_j), \quad (6.4)$$

with $\text{vec}(\mathbf{X})$ as a row-major vectorization of the corresponding assignment matrices and \mathbf{Q}_{ij} as the cost matrix of the frame-pair. The maximum frame gap Δf_{\max} that is modeled in our approach depends only on the density of the cost matrix. To include a connection between frames i and j , the matrix \mathbf{Q}_{ij} needs to be filled with the corresponding similarity scores. The cost matrix \mathbf{Q}_{ij} is sparse, as it also represents all terms that correspond to detection pairs matched to different tracks, which add no cost. Furthermore, no cost is associated with the mapping of a frame to itself, which includes the main diagonal of \mathbf{Q} .

A complete sequence consisting of F frames, can be represented with the stacked assignment matrix

$$\mathbf{z} = [\text{vec}(\mathbf{X}_1)^T, \dots, \text{vec}(\mathbf{X}_F)^T]^T. \quad (6.5)$$

And the corresponding cost

$$c = \sum_{i=1}^F \sum_{j=1}^F c_{ij} = \mathbf{z}^T \mathbf{Q} \mathbf{z}, \quad (6.6)$$

where \mathbf{Q} is a block-matrix made from all \mathbf{Q}_{ij} .

QUBO FORM. To solve the proposed MOT assignment problem with an adiabatic quantum computer it further needs to be represented as a QUBO task with $\{-1, +1\}$ spin states. This consists of two steps, firstly eliminating the constraints and secondly substituting the variables.

1) Constraints are represented using a Lagrangian multiplier λ . As our formulation does not include inequalities, no additional slack variables with corresponding qubits are required. Given the original quadratic program with constraints

$$\arg \min_{\mathbf{z}} \mathbf{z}^T \mathbf{Q} \mathbf{z} + \mathbf{b}^T \mathbf{z} \quad \text{s.t.} \quad \mathbf{G} \mathbf{z} = \mathbf{d}, \quad (6.7)$$

a QUBO can be formulated as

$$\arg \min_{\mathbf{z}} \mathbf{z}^T \mathbf{Q}' \mathbf{z} + \mathbf{b}'^T \mathbf{z} \quad (6.8)$$

with

$$\mathbf{Q}' = \mathbf{Q} + \lambda \mathbf{G}^T \mathbf{G} \quad (6.9)$$

$$\mathbf{b}' = -2\lambda \mathbf{G}^T \mathbf{b}. \quad (6.10)$$

2) Variables are substituted by replacing the optimization variables $z \in \{0, 1\}$ with $s \in \{-1, 1\}$ by using $z = 1/2(s + 1)$ the resulting optimization problem reads

$$\arg \min_{\mathbf{s}} \mathbf{s}^T \mathbf{Q} \mathbf{s} + \mathbf{b}'^T \mathbf{s} \quad \text{with} \quad \mathbf{b}'^T = 2(\mathbf{b}^T + \mathbf{1}^T \mathbf{Q}). \quad (6.11)$$

LAGRANGIAN OPTIMIZATION. Solving the Lagrangian would require solving a problem in both discrete and continuous optimization variables (assignment, and Lagrangian multipliers, respectively). To solve the problem using AQC, we presented a constant penalty reformulation in the previous paragraph, which fixes the Lagrangian multipliers λ . In such an approach, if λ is large enough, constraint satisfaction is guaranteed. More precisely, a quadratic equality constraint reformulation of the form

$$\lambda \|\mathbf{G} \mathbf{z} - \mathbf{d}\|_2^2, \quad (6.12)$$

is used in Equations 6.9 and 6.10, which allows to only consider positive Lagrangian multipliers λ . Even though λ needs to be just large enough from a theoretical perspective, in practice it should be as small as possible. This is especially relevant for AQC, as with a high λ the conditioning of the corresponding Hamiltonian in the AQC gets worse. This should be avoided as it results in a lower probability of finding the correct solution in each measurement.

Thus, in practice a problem dependent bound for the minimum penalty term λ_{\min} should be used. One approach to reduce the spectral gap is to

estimate an individual λ_i for each constraint $\mathbf{G}_i \mathbf{x} = \mathbf{d}_i$ using upper bounds. While such bounds can be computed, they are not tight in many cases. We, therefore, propose a heuristic to estimate the Lagrangian multipliers λ_i that closely match their minimal value $\lambda_{i,\min}$. Each multiplier is modeled by

$$\lambda_i = \lambda_b + \lambda'_i + \lambda_{\text{off}}, \quad (6.13)$$

where λ_b is a small base value that resolves the easy to fulfill constraints, λ'_i is estimated during the optimization procedure and λ_{off} is an offset to increase the spectral gap.

Starting with $\lambda'_i = 0$ and $\lambda_{\text{off}} = 0$ for all constraints, the QUBO is solved using annealing. In general, this will result in a solution \mathbf{z}_λ that does not fulfill the constraints. As in our formulation, only positive violations result in a cost improvement, i.e. $\mathbf{Gz} \geq \mathbf{d}$, the cost reduction of a constraint violation can be estimated as

$$a_i(\mathbf{z}_\lambda) = 2(\mathbf{z}_G^T \mathbf{Qz}_\lambda - \min_j \mathbf{z}_{G_j}^T \mathbf{Qz}_\lambda) / v_i^2, \quad (6.14)$$

$$\mathbf{z}_G^T = (\mathbf{G}_i \circ \mathbf{z}_\lambda^T) \quad (6.15)$$

$$v_i(\mathbf{z}_\lambda^0) = \mathbf{G}_i \mathbf{z}_\lambda - \mathbf{d}_i, \quad (6.16)$$

where \mathbf{z}_G are the variables masked with \mathbf{G}_i and v_i is the degree of violation. To fulfill the corresponding constraint, we set

$$\lambda'_i(\mathbf{z}_\lambda) = -a_i(\mathbf{z}_\lambda) - \lambda_b + \epsilon, \quad (6.17)$$

with a small ϵ to assert that constraint i is fulfilled in the current setting. While this can be evaluated for all constraints simultaneously, the full procedure needs to be performed iteratively, as not all constraints may be violated in the optimal solution. Nevertheless, the set of measurements returned by the AQC can be used to reduce the number of required iterations. Instead of taking a single best solution, all solutions \mathbf{z}_j that are close to the optimal solution are evaluated and merged as $\lambda'_i = \max_j \lambda'_i(\mathbf{z}_j)$. In our formulation, these can be solutions where the track order is permuted.

After estimating the Lagrangian multipliers, the total cost matrix scale is small, nevertheless, the same also holds for the spectral gap, as the cost of not fulfilling constraints is small. Therefore, the additional offset λ_{off} is added to the Lagrangian multipliers.

SIMILARITY COST. We use the same approach for cost generation as AP-lift [96], where multi-layer Perceptrons are used to regress the similarity

score between pairs of detections. Features used to compute this score are the IoU of aligned boxes and the dot-product between DG-Net [258] appearance features. DG-Net features are generated with the network trained on the MOT15 dataset [134] together with [192, 226, 257]. To generate the MLP input vector, the features are normalized with a global context [95], which results in a total of 22 features [96]. Furthermore, assigning the dummy-detection to a track incurs no cost and assigning a detection to the dummy-track, i.e. labeling it as a false-positive, corresponds to a small negative value β . This is required to prevent the assignment of single detections to tracks.

POST PROCESSING. Even in an offline setting, long sequences cannot be represented as a single optimization problem and need to be split into a set of overlapping subproblems. We set the overlap to the modeled frame gap, and match tracks using the common frames. Matching is stated as a linear sum problem that maximizes the number of detections that are jointly assigned to tracks in both subproblems. As multiple subsequent tracks can be modeled by a single track ID, tracks that are interrupted longer than the maximum modeled frame gap Δf_{\max} are separated.

PROBLEM SCALING. One important aspect when designing algorithms for current and near-future quantum computers is the required number of qubits. Many current formulations of MOT grow quickly in size w.r.t. the number of detections, tracks, frames and the length of the modeled frame gap. In contrast to this, the number of qubits in our approach only grows linearly in the number of detections, tracks and frames. Furthermore, by using a quadratic optimization problem, longer frame gaps can be modeled by additional entries in the cost matrix, which correspond to additional couplings between qubits.

While on short sequences the number of possible tracks needs to be at least as high as the total number of tracks, long sequences can profit from a saturation of the required number of tracks. After a track has terminated, there is no cost associated with assigning new detections if they have a distance of more than the maximal frame gap Δf_{\max} from the previous track. Therefore, multiple subsequent real tracks can be modeled by a single track ID and easily be separated in post-processing.

6.3 TRADITIONAL SOLVERS

While our formulation is advantageous when solved on an adiabatic quantum computer, publicly available real systems have not yet reached a scale where large experiments can be performed. We, therefore, use classical solvers to show the results of our approach on real-world tasks, even though a quadratic problem formulation is known to be hard in this context. A common requirement of solvers to perform quadratic binary optimization via branch and bound is the convexity of the continuous relaxation of the problem. This corresponds to a positive-definite cost matrix \mathbf{Q} , i.e. a matrix with only positive eigenvalues, which is not fulfilled for the given cost matrix in most cases.

6.3.1 Hessian Regularization

A common approach to enforce positive eigenvalues is adding an identity matrix scaled by ϵ . As this changes the cost function and thus the optimal solution, small values need to be used for ϵ , making this approach only suitable for compensating small negative eigenvalues. Nevertheless, investigating the constraints of our formulation leads to a sparse diagonal matrix \mathbf{E} that can be added to the cost matrix \mathbf{Q} without changing the optimal solution. With the same approach of grouping the total cost matrix into blocks between frames as in Equation (6.6), the following definition of \mathbf{E} is provided in blocks between frames. As only diagonal elements are relevant, blocks between different frames are zero matrices $\mathbf{E}_{ij} = \mathbf{0} | i \neq j$. The blocks on the diagonal, which represent the mapping of a frame i to itself \mathbf{E}_{ii} are diagonal matrices defined by the diagonal elements

$$e_{idt} = \begin{cases} e & d \in \{1, \dots, D\}, t \in \{1, \dots, T-1\} \\ 0 & t = T \end{cases}. \quad (6.18)$$

The indices refer to the position on the diagonal that correspond to detection d and track t . Given a block's assignment matrix \mathbf{X}_i , the total cost of the block after adding the diagonal term is

$$c_{ii} = \text{vec}(\mathbf{X}_i)^T (\mathbf{Q}_{ii} + \mathbf{E}_{ii}) \text{vec}(\mathbf{X}_i) = e(T-1), \quad (6.19)$$

with $\mathbf{Q}_{ii} = \mathbf{0}$ and T tracks in total. The intuition and proof behind the definition is given in the following.

Given a binary problem, any diagonal entry adds cost if a variable is active. In the detection track assignment problem, this corresponds to adding a constant if a detection is assigned to a track. As constraint 6.1 asserts that exactly one detection (real- or dummy-detection) is assigned to every real track each time-step, having a cost e for the assignment adds this cost for each of the $T - 1$ real tracks. As the constraint does not apply for the dummy-track with index T and an arbitrary number of detections may be assigned to it. Therefore, the same argument would not hold and we can not add an additional cost to these entries ($e_{ikl} = 0 | t = T$), without influencing the total cost function. The proof for this is as follows and holds given a binary optimization problem $x \in \{0, 1\}$ and the constraints in Equations (6.1) and (6.2).

$$\begin{aligned}
 c_{ii} &= \text{vct}(\mathbf{X}_i^T) \mathbf{E}_{ii} \text{vct}(\mathbf{X}_i) = \text{diag} \sum_{t=1}^T \sum_{d=1}^D x_{id,t}^2 e_{d,t} = \text{bin} \sum_{t=1}^T \sum_{d=1}^D x_{id,t} e_{d,t} \\
 &= \sum_{t=1}^{T-1} \sum_{d=1}^D x_{id,t} e_{d,t} + \sum_{d=1}^D x_{id,T} e_{d,T} \stackrel{(22)}{=} \sum_{t=1}^{T-1} \sum_{d=1}^D x_{id,t} e + \sum_{d=1}^D x_{id,T} 0 \quad (6.20) \\
 &= e \sum_{t=1}^{T-1} \sum_{d=1}^D x_{id,t} \stackrel{(6)}{=} e \sum_{t=1}^{T-1} 1 = e(T - 1)
 \end{aligned}$$

6.3.2 Post Processing

To allow the handling of long sequences that cannot be represented as a single optimization problem, the sequence needs to be split into overlapping subproblems. We split a long sequence in equally sized subproblems with an overlap similar to the modeled frame gap. After tracking each subproblem separately, tracks are matched between each pair of neighboring subproblems by solving a linear sum problem that can be solved in polynomial time. The optimization goal is to maximize the number of detections that are jointly assigned to tracks matched in both subproblems. The linear sum optimization problem for matching subproblems k and $k + 1$ is stated as

$$\begin{aligned}
 \max_{x_{ij} \in \{0,1\}} \sum_{i=1}^{T_k} \sum_{j=1}^{T_{k+1}} x_{ij} m_{ij} \quad \text{s.t.} \quad & \sum_{i=1}^{T_k} x_{ij} \leq 1 \\
 & \sum_{j=1}^{T_{k+1}} x_{ij} \leq 1,
 \end{aligned} \quad (6.21)$$

where x_{ij} are the optimization variables indicating an assignment of track i in segment k to track j in segment $k + 1$, The considered tracks T_k and T_{k+1}

are the tracks that have at least one detection assigned to them in the frames overlapping between both subproblems. m_{ij} is the number of detections shared by tracks i and j in the overlapping frames, which furthermore is set to a small negative value if tracks i and j have no overlap.

6.4 EXPERIMENTS AND RESULTS

AQC experiments are performed on a D-wave Advantage 4.1 [160]. The system contains at least 5000 qubits and 35,000 couplers implemented as superconducting qubits [29] and Josephson-junctions [86] respectively. Every qubit of the D-wave Advantage is connected to 15 other qubits, which needs to be reflected in the sparsity pattern of the cost matrix. If a denser matrix is required, chains of qubits are formed that represent a single state. The actual parameters can vary due to defective qubits and couplers. All experiments are performed using an annealing time of $1600 \mu\text{s}$ and an additional delay between measurements to reduce the inter-sample correlation. In the following, a single measurement refers to the combination of an annealing cycle and the subsequent measurement.

Simulated annealing is used to evaluate our approach in a noise-free setting. We use the simulation provided by D-wave for this purpose.

Classical solvers are used to demonstrate the performance of the proposed algorithm on the full MOT15 dataset. All experiments using classical solvers are performed using Gurobi [83] with CVXPY [57] as a modeling language.

6.4.1 Lagrangian Multiplier

Fixed Lagrangian multipliers represent the basic approach to include constraints in the QUBO. We run experiments with synthetic tracking sequences where object detections are in random order. The scenarios are defined by their similarity scores, which we set to 0.8 for a match and -0.8 for different objects. Furthermore, we add Gaussian noise with variance σ^2 to the similarity scores and subsequently truncate them to $[-1, 1]$. In the experiments 3 detections over 5 frames and a noise level between $\sigma = 0.2$ to $\sigma = 1.0$ is used. The tracking parameters are set to 4 tracks and a maximal frame-gap of $\Delta f_{\max} = 3$ frames.

Results generated with simulated annealing are shown in Figure 6.2, where the top plot shows the solution probability for different noise levels over an increasing Lagrangian multiplier. For each λ , 4096 measurements are performed. The lower plot shows the histogram over the energy of the

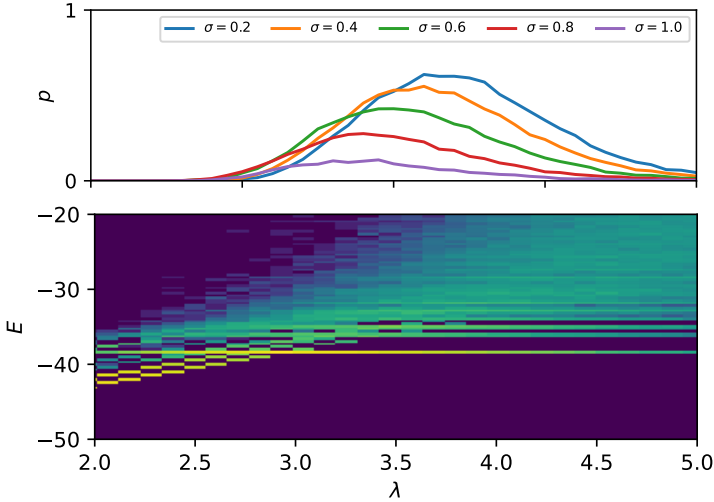


FIGURE 6.2: Solution probability and energy levels using simulated annealing for different noise levels and changing λ .

returned solutions for a noise level of $\sigma = 0.6$. The correct solution can be seen at an energy level of -38.6 .

With increasing noise level, the solution probability for the best value of λ reduces considerably, which can be explained by the energy histogram. As described in Section III.II, a low spectral gap, i.e. the difference between the lowest and second-lowest energy level, reduces the probability of the AQC staying in its ground state and thus, the probability of finding the correct solution. In the energy plot, the spectral gap is visible as the distance between the energy band of the correct solution and the next higher energy band, given a sufficiently high λ , such that the correct solution has the lowest energy.

Tracking with the D-wave advantage is performed on a problem with 3 detections over 4 frames and noise levels $\sigma \in \{0.0, 0.1, 0.2\}$. Results using 4000 measurements for each setting are shown in Figure 6.3. Solution probabilities are lower compared to simulated annealing and high energy solutions are returned more often. This can be explained by the high noise of current AQCs.

OPTIMIZED LAGRANGIAN multipliers are introduced to improve the spectral gap of the normalized cost matrix. We perform the same tracking

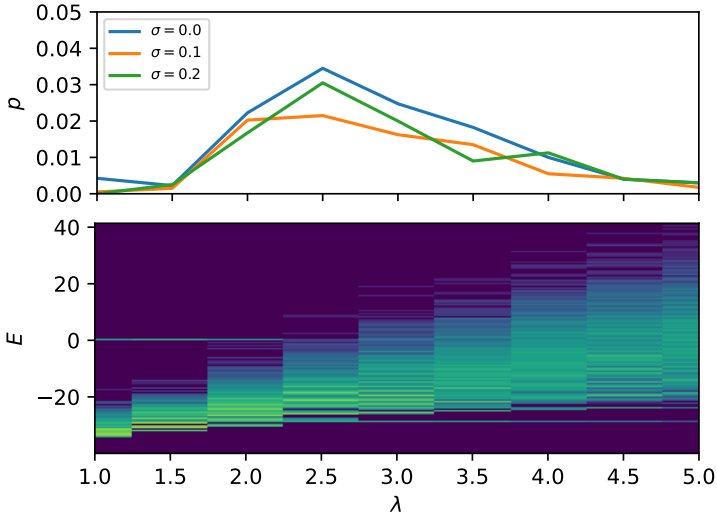


FIGURE 6.3: Solution probability and energy levels using quantum annealing for different noise levels and changing λ .

tasks as for fixed Lagrangian multipliers, but evaluate the results w.r.t. the offset term λ_{off} . Results generated with simulated annealing are shown in Figure 6.4. Optimization of the Lagrangian multipliers is initialized with a base value of $\lambda_b = 0.5$. The probability of finding the right solution is increased and stays high over a large range of λ_{off} compared to only using a single λ . Furthermore, the best solution probability for each of the noise levels is better than the optimum for a fixed Lagrangian multiplier. This has two advantages: first, fewer measurements are needed to find the correct solution and secondly, less effort needs to be invested to find a good setting for λ . Results for the problem with an optimized Lagrangian multiplier with $\lambda_b = 1.0$ solved on the AQC are shown in Figure 6.5. When optimally tuned for $\sigma = 0$, our method returns the best solution in 4.8% of the measurements, compared to 3.5% when using a fixed multiplier. Furthermore, even without an additional offset $\lambda_{\text{off}} = 0$, the best solution is returned in 0.8% of the measurements.

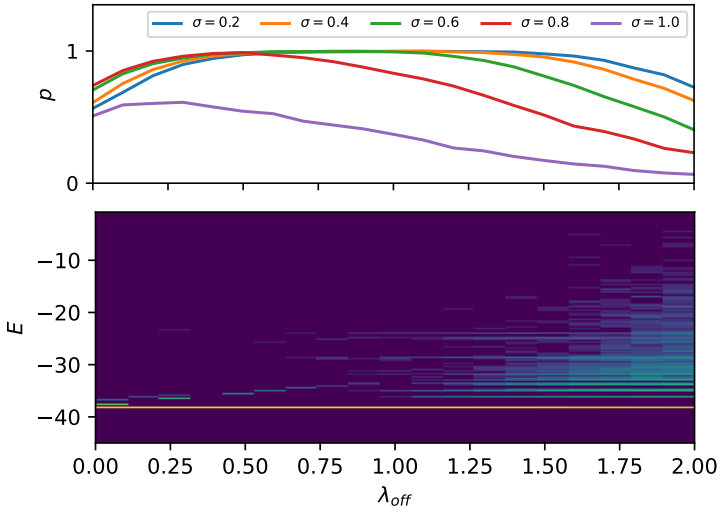


FIGURE 6.4: Solution probability and energy levels using simulated annealing and optimized λ_i for different noise levels over λ_{off} .

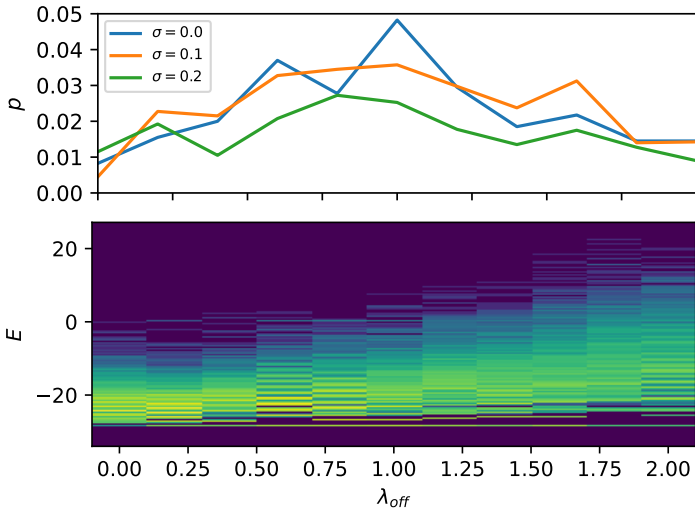


FIGURE 6.5: Solution probability and energy levels using quantum annealing and optimized λ_i for different noise levels over λ_{off} .

Method		MOTA	IDF ₁	MT	ML	FP	FN	IDs
Test	Lif_T [95]	52.5	60.0	244	186	6837	21610	730
	MPNTrack [28]	51.5	58.6	225	187	7260	21780	375
	ApLift [96]	51.1	59.0	284	163	10070	19288	677
	MFL_TST [242]	49.2	52.4	210	176	8707	21594	912
	Tracktor [18]	44.1	46.7	130	189	6477	26577	1318
	Ours	49.9	53.5	187	179	5924	24032	1689
X-val	AP lift	59.6	67.8	237	133	8897	10150	283
	Ours	59.7	67.6	234	134	8720	10214	370

TABLE 6.1: Results on MOT15 [134]. X-val refers to results on the training set using leave-one-out cross validation.

6.4.2 MOT15

We use the MOT15 dataset [134] to show that our method performs on par with state-of-the-art tracking methods and present results in Table 6.1. For this dataset, GUROBI [83] is used to find a solution for the optimization problem. The sequence is evaluated in segments of 20 frames using a maximum frame gap of $\Delta f_{\max} = 10$. As binary quadratic problems are very hard to solve with classical approaches, it is not possible to find an optimum solution for segments that contain a high number of tracks. In these cases, we terminate the optimization after 900s on a single segment and use the best solution found.

For comparisons, ApLift [96] is closest to our method, as it uses the same set of similarity features. On the test set, we achieve a MOTA-score of 49.9% and perform only 1.2% below ApLift, even though it models gaps up to 50 frames.

For a comparison under similar settings, we evaluate our method and ApLift [96] with the same frame gap of $\Delta f_{\max} = 10$. As MOT15 does not contain a validation set, we use leave one out cross-validation on all samples of the training set for a fair comparison. In this scenario, our method improves by 0.2% over ApLift in MOTA score. An explanation for this is that the MOT15 test set contains more detections in each frame on average (10.6 vs. 7.3) than the training set. In this case, there are more sequences where the classical solver does not find a solution and thus, generates a non-optimal result.

Further detailed results for our method on each sequence in the MOT15 [134] training and test set are provided in Table 6.2. While the results on both sets

are competitive with current state-of-the-art methods [96], the performance on the training set with leave one out cross-validation is higher than on the test set.

The difference can be explained by the harder examples represented by it. While both splits contain a similar number of frames (5500 frames and 5783 frames respectively), the number of tracks, detected boxes and the corresponding density is approximately 45% higher in the test set and thus, also the complexity and size of the optimization problem. As our formulation is designed for AQC using an Ising model, the resulting optimization problem is a quadratic binary program and thus, hard to solve on classical hardware. This becomes apparent for two sequences in the test set, *AVG-TownCentre* and *PETS09-S2L2* with a high density of 15.9 and 22.1 and low frame rate of 2.5 fps and 7 fps respectively. The two sequences account for only 27.3% of the total detections, but for 58.7% of the ID switches. ID switches are a good measure for the tracker’s performance in this case, as they are less influenced by the performance of the object detector than FP and FN. Due to the larger size of these problems, the optimization cannot finish for all segments within the given time frame and thus, returns a sub-optimal solution.

Even though the problem size is a limitation when solving the problem on classical hardware, it can be resolved when future AQCs become available. As the overall performance is similar to current state-of-the-art methods on MOT15 [134], it can be expected that it scales up to larger datasets accordingly and thus, provides the basis to develop AQC based formulations of the MOT task.

MOT15 with AQC. To show that tracking with an AQC already scales to small real-world examples, a part of the *PETS09-S2L1* sequence is used. As the problem size has to be limited, three tracks that contain two occlusions, are extracted between frames 121 to 155. We execute our pipeline on segments of 5 frames with 3 tracks, a maximum frame-gap of 3, and optimized Lagrangian multipliers. The subproblems are solved on the D-wave Advantage with 1600 μs annealing time and 500 measurements per segment. The most relevant frames that highlight occlusions are shown in Figure 6.6. The normalized energy $E - E_0$ levels of the measurements for each subproblem are shown at the top of Figure 6.7 and the corresponding probabilities p of measuring the right solutions are plotted in the lower one. The subproblems 5 and 10 correspond to the two occlusions highlighted in Figure 6.6. These are harder to solve problems, as multiple solutions with

seq	MOTA	IDF ₁	MT	ML	FP	FN	IDs	Density	Tracks	Boxes	FPS
Venice-2	41.6	50.0	13	1	2178	1855	135	11.9	26	7141	30
KITTI-17	79.6	83.6	6	0	5	130	4	4.7	9	683	10
KITTI-13	33.5	57.8	13	11	197	293	17	2.2	42	762	10
ADL-Rundle-8	26.7	51.4	18	3	3587	1336	49	10.4	28	6783	30
ADL-Rundle-6	63.3	53.7	11	1	228	1570	40	9.5	24	5009	30
ETH-Pedcross2	46.2	59.9	28	74	127	3216	27	7.5	133	6263	14
ETH-Sunnyday	78.1	87.0	19	6	110	295	2	5.2	30	1858	14
ETH-Bahnhof	47.3	67.5	98	38	1933	895	24	5.4	171	5415	14
PETS09-S2L1	83.2	76.9	17	0	341	351	58	5.6	19	4476	7
TUD-Campus	75.5	75.4	4	0	9	72	7	5.1	8	359	25
TUD-Stadtmitte	81.6	80.8	7	0	5	201	7	6.	10	1156	25
OVERALL	59.7	67.6	234	134	8720	10214	370	7.3	500	39905	-
Venice-1	44.4	49.0	6	3	656	1839	42	10.1	17	4563	30
KITTI-19	48.2	60.1	14	17	528	2191	49	5.0	62	5343	10
KITTI-16	52.7	67.1	3	1	120	666	19	8.1	17	1701	10
ADL-Rundle-3	50.0	47.4	10	7	653	4346	81	16.3	44	10166	30
ADL-Rundle-1	38.2	49.9	12	2	2365	3313	73	18.6	32	9306	30
AVG-TownCentre	52.7	57.0	58	35	363	2767	250	15.9	226	7148	2.5
ETH-Crossing	62.3	75.1	7	8	38	335	5	4.6	26	1003	14
ETH-Linthescher	56.5	62.3	45	89	342	3493	48	7.5	197	8930	14
ETH-Jelmoli	51.0	65.5	18	13	522	701	19	5.8	45	2537	14
PETS09-S2L2	50.1	38.7	2	4	312	4259	243	22.1	42	9641	7
TUD-Crossing	85.7	81.6	12	0	25	122	11	5.5	13	1102	25
OVERALL	49.9	53.5	187	179	5924	24032	840	10.6	721	61440	-

TABLE 6.2: Results of our method on the MOT₁₅ [134] training and test set. Results on the training set are generated using leave-one-out cross validation (X-Val).

small differences in their energy exist and thus, they have a lower solution probability.

6.5 CONCLUSION

In this chapter, we proposed the first quantum computing formulation of MOT. We demonstrated that current AQC's can solve small real-world tracking problems, and that our approach closely matches state-of-the-art MOT methods. Current limitations stem from the proposed formulation being optimized to run on an AQC. As QUBO is known to be hard using classical approaches and as current AQC's are still at an experimental stage, problems are limited to a small scale. Nevertheless, quantum computing has the potential to make much larger problems feasible in the future.



FIGURE 6.6: Frames from the extracted sequence tracked on the AQC.

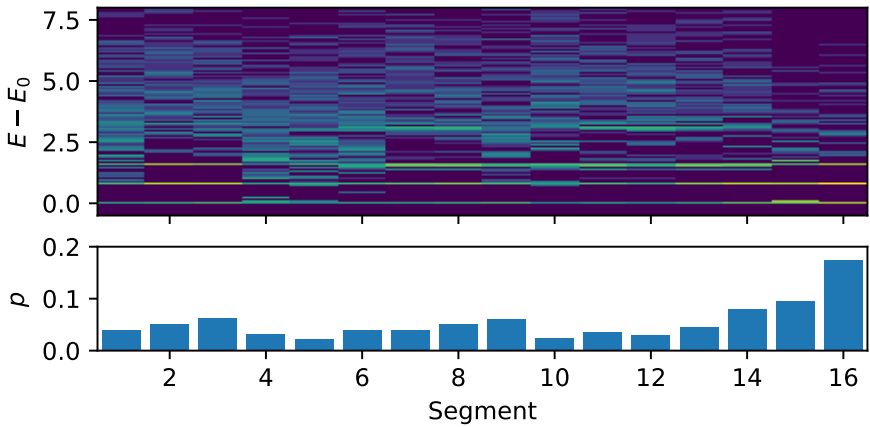


FIGURE 6.7: Energy of measurements returned by performing tracking of the PETS09-S2L1 sequence on the D-wave Advantage. The bar-plot shows the probability of measuring the optimal solution.

PROBABILISTIC SAMPLING OF BALANCED K-MEANS USING ADIABATIC QUANTUM COMPUTING

AQC is a promising quantum computing approach for discrete and often NP-hard optimization problems. Current AQCs allow to implement problems of research interest, which has sparked the development of quantum representations for many computer vision tasks. Despite requiring multiple measurements from the noisy AQC, current approaches only utilize the best measurement, discarding information contained in the remaining ones.

To use the AQC more efficiently, we explore the potential of using this information for probabilistic balanced k-means clustering. Instead of discarding non-optimal solutions, we propose to use them to compute calibrated posterior probabilities with little additional compute cost. This allows us to identify ambiguous solutions and data points, which we demonstrate on a D-Wave AQC on synthetic and real data.

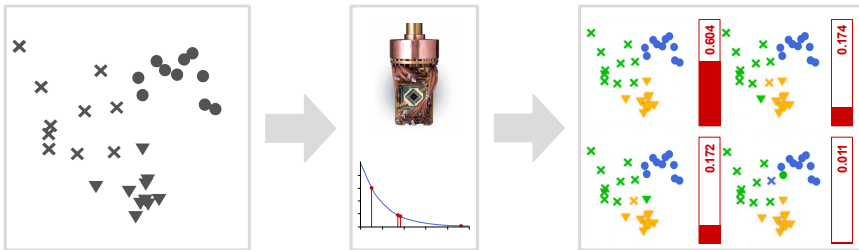


FIGURE 7.1: The proposed approach uses an adiabatic quantum computer to sample solutions of a balanced k-means problem. By using an energy-based formulation, likely solutions are drawn from a Boltzmann distribution. By reparametrizing the distribution, the calibrated posterior probability of each solution can be estimated.

7.1 INTRODUCTION

Clustering is a fundamental problem in machine learning and computer vision, extensively employed for the analysis and organization of large volumes of unlabeled visual data. It involves grouping similar objects based

on their features, facilitating efficient data processing and unsupervised learning algorithms [238]. The information extracted from clusters plays a vital role in various computer vision applications, including image classification [33, 131], segmentation [46, 56], tracking [117], and network training [45, 241]. However, the often NP-hard computational complexity of solving clustering problems often hinders their application to large-scale problems that require fast processing times [238].

In addition to this, the ambiguity of the task itself contributes to the absence of a unique definition for clustering [238], which arises from the range of possible optimization objectives. An example of this are different algorithms that can optimize for the most compact solution, an underlying data-generating process or just use local distances. One way to address this challenge is the use of confidence-based estimators and the sampling of multiple likely solutions. This approach helps in identifying well-assigned samples and uncovering ambiguities within the data [238].

Quantum algorithms that promise a considerable speedup over their classical counterparts and are inherently probabilistic have the potential to enable a new family of machine learning algorithms. By now, quantum machine learning algorithms approach tasks such as optimization of quadratic problems [111], training of restricted Boltzmann machines [58], and learning with quantum neural networks [1]. While the current quantum computers can only solve small-scale problems, they provide the basis to develop and test algorithms that can considerably increase the size of feasible problems in the future.

In our framework, we propose to exploit the probabilistic nature of a quantum computer to sample multiple high-probability solutions of the balanced k-means problem at little additional cost. By formulating the k-means objective as a quadratic energy function, we embed the clustering task into the quantum-physical system of an AQC. By repeatedly measuring the quantum system, we sample high-probability solutions according to the Boltzmann distribution. Unlike previous approaches that only use the best solution and discard all other measurements, we utilize all samples to generate probabilistic solutions for the k-means problem, as shown in Figure 7.1. We recalibrate the samples of the clustering problem to address temperature mismatch [183], estimating the posterior probability for each solution, which allows us to identify ambiguous points and provides alternative solutions. We demonstrate the algorithm on a D-Wave quantum computer and perform extensive experiments in simulation. However, our primary objective is to explore the potential and efficient utilization of

quantum computing for machine learning. Our primary contributions are as follows:

- We propose a quantum computing formulation of balanced k-means clustering that predicts well-calibrated confidence values and provides a set of alternative clustering solutions.
- A reparametrization approach is used to compute posterior probabilities from samples that avoids exact tuning of the AQC sampling temperature.
- Extensive experiments on synthetic and real data show the calibration of our approach using simulation as well as the D-Wave Advantage 2 AQC prototype.

7.2 THEORY

In this section, the theory on energy-based models as well as the corresponding clustering background are introduced. For an overview of the quantum computing background required for the remainder of this chapter the reader is referred to Section III.II

7.2.1 Energy-Based Models

Energy-based models are probabilistic models that use an energy function $E(\mathbf{x})$ to describe the probability of each system state. By assigning an energy value to each system state, the probabilities are Boltzmann distributed

$$p(\mathbf{x}) = \frac{1}{A} \exp[E(\mathbf{x})/T], \quad (7.1)$$

with the temperature T as a free parameter in this model and the normalization constant A ensuring that the values are valid probabilities. Our model follows this approach and represents the clustering problem as an energy-based formulation.

One challenging task to use energy-based models remains in sampling the system according to the Boltzmann distribution as this is NP-hard for the quadratic Ising-model formulation [183]. While this is usually approximated on classical hardware with the corresponding computational overhead, AQC provides a direct way to sample from a physical system that follows the Boltzmann distribution [183].

7.2.2 Clustering

Clustering is one core task required in many unsupervised machine learning algorithms. It has the objective to group a set of points \mathbf{X} into disjoint clusters $\{c_1, \dots, c_K\}$, where each cluster contains points that are similar in the feature space. A design parameter is the definition of the similarity metric, which leads to a wide range of clustering algorithms available. The popular and simple k -means algorithm [71, 148] optimizes the quadratic distance to a centroid. Such distance-based metrics assume clusters to be compact, while other approaches such as *dbscan* [65] define features as similar if they come from a contiguous region of high density.

In our formulation, balanced clustering is investigated, where a defined target cluster size s_k is required. This approach forms the basis of many AQC-based algorithms [17, 23, 250] and has a wide range of further applications such as secret key generation in cryptography [93], energy-efficient data aggregation in wireless sensor networks [142], or data cleaning [69].

Uncertainty estimation in clustering and machine learning aims at quantifying the confidence that predictions made by a model correspond to the underlying ground-truth. The confidence scores can be used to eliminate uncertain samples from the solution, to select a set of possible hypotheses or to find the right parameters of the algorithm itself [36].

Uncertainty estimates can be generated separately for each data-point by evaluating the likelihood function $p(\mathbf{X}|\hat{\mathbf{Z}})$, or by computing the posterior probability $p(\hat{\mathbf{Z}}|\mathbf{X})$ that describes the probability of the whole clustering solution. As the former can be evaluated if the data generating model is defined, it can be combined with most clustering approaches. Nevertheless, as the likelihood function is not a probability distribution of the cluster assignments, it does not provide a calibrated prediction. In contrast to this, the posterior probability $p(\mathbf{Z}|\mathbf{X})$ directly represents the probability that the estimated assignment \mathbf{Z} corresponds to the ground-truth. While this provides an interpretable result, it is often infeasible to compute as all possible assignments need to be evaluated in order to find the normalizing constant.

The set of possible clustering solutions together with their posterior probabilities provides valuable information that can be utilized for a range of low- as well as high-level reasoning tasks. On a low level, the probability of the best solutions can be used to identify the right number of clusters. If a number different than the number of the data-generating process is chosen, additional ambiguity is introduced. For a larger number during clustering

than present in the data, the strong overlap between the corresponding distributions induces high ambiguity, while an insufficient number of clusters spreads points inside a single cluster far apart, which reduces the associated probability.

For high-level applications, calibrated samples of possible clustering solutions can provide additional information in multi-object tracking [42]. Following the AQC framework, tracking can be implemented as a clustering problem with additional constraints that represent the temporal relation between points [250]. Each cluster then corresponds to one object in the video, with each point representing it at a different timestep. The feature used to define clusters can either be visual similarity from re-identification [90, 176, 258] features or spatial similarity [20]. Different solutions thus represent different tracks through time, where ambiguities can be generated by occlusions or crossing paths. In a larger system such as an autonomous vehicle, knowing all likely candidate solutions can allow for predicting multimodal candidates for future trajectories [55, 108] and can help to evaluate the risk of taking any action. In a similar approach, feature matching can be formulated [23] and ambiguous candidates can be discarded during 3d reconstruction or camera pose estimation.

7.2.3 Clustering as QUBO

To solve the clustering problem using AQC, a QUBO formulation is required. We use a variation of the one-hot encoding approach [11, 50] that uses a matrix $Z \in \{0, 1\}^{K \times I}$ to encode the cluster assignment of I samples to K clusters. Each row corresponds to one of K clusters and each column to one of I samples. An entry $Z_{ki} = 1$ indicates that the sample x_i belongs to cluster c_k .

As each sample needs to be assigned to a single cluster, the sum of each column of Z needs to satisfy the constraint $\sum_k Z_{ki} = 1 \forall i$. The implementation of constrained clustering, where each cluster has a fixed size s_k furthermore requires the row constraints $\sum_i Z_{ki} = s_k \forall k$ on Z .

With the cost $q(i, j, k)$ for assigning the pair of samples x_i and x_j to the same cluster c_k , the optimization problem reads

$$\begin{aligned} \hat{Z} &= \arg \min_Z \sum_k \sum_i \sum_j Z_{ki} Z_{kj} q(i, j, k) \\ \text{s.t.} \quad &\sum_k Z_{ki} = 1 \forall i \quad \wedge \quad \sum_i Z_{ki} = s_k \forall k. \end{aligned} \tag{7.2}$$

By vectorizing Z in row-major order as $\mathbf{z} = \text{vec}(Z)$, the optimization problem can be rewritten in matrix form as

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z}} \mathbf{z}^T \mathbf{Q} \mathbf{z} \quad \text{s.t.} \quad \mathbf{G} \mathbf{z} = \mathbf{d}, \quad (7.3)$$

where \mathbf{Q} is a block diagonal matrix with blocks $\mathbf{Q}_0, \dots, \mathbf{Q}_K$ and $\mathbf{G} \mathbf{z} = \mathbf{d}$ corresponds to the matrix formulation of the constraints. Each block \mathbf{Q}_k of the cost matrix is a square matrix that contains the costs $q(i, j, k)$ at $\mathbf{Q}_{k,ij}$.

While the constraints are not directly covered by the QUBO problem, they can be modeled using Lagrangian multipliers. To avoid a mixed discrete and continuous optimization problem, a quadratic penalty reformulation $\lambda \|\mathbf{G} \mathbf{x} - \mathbf{d}\|_2^2$ is chosen, leading to the minimization problem

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z}} \mathbf{z}^T \mathbf{Q}' \mathbf{z} + \mathbf{b}'^T \mathbf{z} \quad (7.4)$$

with

$$\mathbf{Q}' = \mathbf{Q} + \lambda \mathbf{G}^T \mathbf{G} \quad (7.5)$$

$$\mathbf{b}' = -2\lambda \mathbf{G}^T \mathbf{d}. \quad (7.6)$$

In contrast to a linear penalty approach, where the multiplier λ needs to be optimized, our selection only requires a sufficiently high λ , as all constraint violations result in an increased penalty term. With such selection, the penalty term evaluates to zero $\lambda \|\mathbf{G} \mathbf{x} - \mathbf{d}\|_2^2 = 0$ if all constraints are fulfilled, and thus, the minimizer $\hat{\mathbf{z}}$ of the modified optimization problem is a minimizer of the original optimization problem in Equation (7.3).

Besides modeling balanced clustering, further constraints can be introduced using Lagrangian multipliers. This allows to represent a wide range of tasks as QUBO clustering problems solvable with AQC.

7.3 PROBABILISTIC QUANTUM CLUSTERING

7.3.1 Motivation

Our clustering formulation employs a mixture of Gaussians to explain the observations, where the cluster assignments are latent variables. Each sample in a cluster c_k is modeled as a sample drawn from a Gaussian distribution $\mathcal{N}(\bar{\mathbf{c}}_k, \mathbf{I})$ with mean $\bar{\mathbf{c}}_k$ and identity covariance according to the k-means setting. Following a Bayesian approach, the best clustering solution can be found by maximizing the posterior probability over possible assignments Z of points to clusters.

$$p(Z|\mathbf{X}) = \frac{p(\mathbf{X}|Z)p(Z)}{\sum_{Z'} p(\mathbf{X}|Z')} = \frac{p(\mathbf{X}|Z)p(Z)}{A}. \quad (7.7)$$

While this approach provides a probabilistic estimate by jointly modeling information about the possible cluster configurations and the distribution of data points, it is often intractable to evaluate due to the partitioning function $A = \sum_{Z'} p(\mathbf{X}|Z')$. Computing A requires to sum over all possible solutions Z , which grows exponentially with the number of samples in the clustering problem. To overcome this, we utilize an AQC that samples directly from a Boltzmann distribution, which we parameterize according to the probabilistic clustering problem.

7.3.2 Data Model

Determining the cost function of the optimization problem is a design choice of the algorithm. For our probabilistic approach, a well-defined data distribution, that forms the basis of the cost function is required. While many tasks approached in quantum computer vision, such as tracking [250] or synchronization [16, 23], costs are based on learned metrics or heuristics, they can also be trained to reflect the properties required in our approach.

We therefore, follow the mixture of Gaussian model, where each cluster generates samples from a normal distribution. With the independence of observations and clusters, given the distribution parameters, the likelihood of the joint observations for an assignment Z is given as the product of the individual likelihoods

$$f(\mathbf{X}|Z) = \prod_{k=1}^K f(\mathbf{X}|Z_k) = \prod_{k=1}^K \prod_{i \in Z_k} f(\mathbf{x}_i|Z_k) = \quad (7.8)$$

$$\prod_{k=1}^K \prod_{i \in Z_k} \frac{1}{\sqrt{2\pi^d}} \exp \left[-\frac{1}{2} (\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \mathbf{I} (\mathbf{x}_i - \bar{\mathbf{x}}_k) \right] = \quad (7.9)$$

$$\frac{1}{\sqrt{2\pi^d}} \exp \left[-\sum_{k=1}^K \sum_{i \in Z_k} \frac{1}{2} (\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \mathbf{I} (\mathbf{x}_i - \bar{\mathbf{x}}_k) \right] \quad (7.10)$$

where the likelihood $f(\mathbf{x}_i|Z_k)$ corresponds to a Gaussian distribution that follows $\mathcal{N}(\bar{\mathbf{x}}_k, \mathbf{I})$ and d is the dimensionality of the space. This result can be used to formulate the energy-based model with the energy function

$$E(\mathbf{X}|Z) = \sum_{k=1}^K \sum_{i \in Z_k} \frac{1}{2} (\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \mathbf{I} (\mathbf{x}_i - \bar{\mathbf{x}}_k). \quad (7.11)$$

Using the energy function to rewrite the posterior distribution leads to the Boltzmann distribution from Equation 5.14

$$p(Z|\mathbf{X}) = \frac{\exp[-(E(\mathbf{X}|Z) + E(Z))]}{\sum_{Z'} \exp[-E(\mathbf{X}|Z')]} \quad (7.12)$$

In this formulation, the energy of the assignment $E(Z)$ corresponds to the prior $p(Z)$, which models feasible and infeasible solutions by an indicator function. In the case of balanced clustering, this corresponds to allowing assignments that have each point assigned to exactly one cluster and a cluster size according to s_k .

Searching the most likely clustering solution corresponds to finding lowest energy solution on the AQC, as well as to the Maximum-a-Posteriori \hat{Z}_{map} estimate of the assignment

$$\hat{Z}_{\text{map}} = \arg \max_Z p(Z|\mathbf{X})p(Z) = \arg \min_Z E(\mathbf{X}|Z) + E(Z). \quad (7.13)$$

As the AQC qubit system is an Ising model, it requires formulating $E(\mathbf{X}|Z)$ and $E(Z)$ quadratic in the optimization variables Z , enabling the joint discovery of assignments and cluster means. This is achieved by using the maximum likelihood estimator of the mean, resulting in a quadratic energy formulation that fits the Ising model in Equation 7.4

$$E_k(\mathbf{X}|Z) = \frac{1}{s_k} \sum_i \sum_j Z_{ki} Z_{kj} (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j). \quad (7.14)$$

The second energy term $E(Z)$ in Equation 7.12, modeling the prior distribution over possible assignments cannot exactly be embedded in the Ising model. We therefore use the quadratic penalty method as shows in Equation (7.4) to approximate the energy of an indicator function. Importantly, this does not influence the energy of feasible solutions, as $\lambda \|\mathbf{G}\mathbf{x} - \mathbf{d}\|_2^2$ evaluates to zero for all solutions that fulfill the clustering constraints.

7.3.3 Boltzmann Reparametrization

While ideally the measurements on AQC are direct samples from the Boltzmann distribution [47, 58], it requires solving a range of challenges that prohibit a direct use in our scenario [183]: Mapping between the cost function and the physical system implemented on the AQC requires Hamiltonian scaling [183]. Estimating the scaling factor is nontrivial [171] and

prohibitive for using samples directly in many cases. Additionally, hardware limitations including imperfection of the processor and the spin-bath polarization effect [183] prevent the AQC to sample the Boltzmann distribution exactly. Finally, as the energy term $E(Z)$ can only be implemented using the penalty method, the sampling density is influenced by solutions not fulfilling the constraints.

We compensate for these limitations by evaluating the energy of all measured feasible solutions Z' and recompute $p(Z|\mathbf{X})$ by evaluating the partitioning function over these, which only requires sampling solutions at a sufficiently high temperature.

7.3.4 Maximum Pointsets

Having the set of the most likely clustering solutions Z available together with their posterior probability $P(Z|\mathbf{X})$ allows to find an assignment Z^* of a subset of points that solves the clustering problem with increased probability $P(Z^*|\mathbf{X})$. Such a set can be found by using Algorithm 1, which implements a greedy approach that disregards points that disagree between different clustering solutions.

Algorithm 1 MaxsetSearch

```

1:  $Z^* \leftarrow Z_0$ 
2:  $p \leftarrow P(Z_0|\mathbf{X})$ 
3:  $i \leftarrow 1$ 
4: while  $p \leq p_{\min}$  do
5:    $Z'_i \leftarrow \text{align}(Z_i, Z^*)$ 
    $\triangleright$  Find the cluster permutation that minimizes the number of points
   assigned to different clusters in  $Z_i$  and  $Z^*$ .
6:    $Z^* \leftarrow Z^* \cap Z'_i$ 
    $\triangleright$  Remove points assigned to different clusters.
7:    $p \leftarrow p + P(Z_i|\mathbf{X})$ 
8:    $i \leftarrow i + 1$ 
9: end while
10: return  $Z^*$ 

```

In the case of a sufficiently well sampled Boltzmann distribution, the resulting maxset assignment Z^* with the corresponding probabilistic estimate $P(Z^*|\mathbf{X})$ is still calibrated.

7.3.5 Inference Parameter Optimization

Using the quadratic penalty method to implement the constraints requires finding suitable Lagrangian multipliers λ . Even though a very high multiplier theoretically guarantees finding a feasible solution, it also deteriorates the conditioning of the optimization problem. Therefore, a suitable Lagrangian multiplier lifts the cost of any constraint violation above all relevant solutions of the clustering problem, while keeping them low enough to avoid scaling the total energy of the problem up considerably. To estimate the multipliers, we follow an iterative procedure as also proposed in [250].

In an initial step, balanced k-means [154] is used to find a feasible clustering solution. This solution is used to estimate Lagrangian multipliers that avoid any first-order violation of the constraints. In subsequent iterative optimization steps, the problem is solved in simulation, to find multipliers that result in a well conditioned problem.

Such optimization procedure is crucial due to the low fidelity of the current generation of AQCs, which requires careful engineering of the problem energy. Therefore, we expect this procedure to become of reduced importance with the progress of quantum computing.

7.4 EXPERIMENTS AND RESULTS

We perform experiments on synthetic data as well as real data to verify the efficacy of our method in finding the set of high-probability solutions and in estimating calibrated confidence scores. The experimental scenarios are solved with QA, Simulated Annealing (SIM), and exact exhaustive search using the presented energy formulation and with k-means as a baseline method, which all optimize for the same cost objective. This further allows us to understand the limitations and required work when deploying the approach to real quantum computers.

7.4.1 Solver Methods

Quantum annealing (QA) experiments are performed on the D-Wave Advantage 2 Prototype 1.1 [159]. The system offers 563 working qubits, each connected with up to 20 neighbors. For each clustering problem 5000 measurements are performed, each with $50\mu\text{s}$ annealing time.

Due to the strong compute-time limitations on an AQC, all Lagrangian optimization steps are performed with SIM, before measuring the final results on the AQC.

Simulated annealing (SIM) provided by D-Wave is used for larger scale comparisons. Similar to QA, we perform 5000 runs for each clustering problem. We reduce the number of sweeps performed in each run to 30, which allows us to sample the Boltzmann distribution at a sufficiently high temperature in most scenarios, making it comparable to QA.

Exact exhaustive search is used as a reference method to validate the energy-based formulation on small problems. By iterating all feasible solutions, the lowest energy solution is guaranteed to be found and the partitioning function is computed exactly.

K-Means clustering with a balanced cluster constraint [154] forms the baseline for our approach. We run the algorithm until convergence for a maximum of 1000 iterations. While this solution does not provide a probabilistic estimate, it is useful to assess the relative clustering performance.

7.4.2 Dataset and Metrics

Data for the quantitative evaluation of our method is synthetically generated. For each clustering problem a total of I points are sampled from a separate normal distribution for each of K clusters. The cluster centers are randomly drawn, such that the distance between each pair of clusters lies within a predefined range $[d_{\min}, d_{\max}]$. For each experiment a total of L clustering tasks is generated. This allows us to evaluate the calibration metrics over a large value range. For all experiments that directly compare methods, identical clustering tasks are used.

Further qualitative examples are provided for the IRIS dataset [70], which contains 50 samples of 4 features in 3 classes. We randomly subsample the points and dimensions to generate the parameters required for our experiments.

Clustering metrics are computed using the available ground-truth clusters. We evaluate 4 standard metrics: The accuracy, which measures the ratio of clustering solutions that are identical to the ground-truth. Completeness [194] measures the ratio of points from a single cluster being grouped together. The adjusted Rand score [102] compares all pairs of points in the ground truth and prediction and the Fowlkes-Mallows index [72] combines precision and recall into a single score.

Solver Method	Accuracy	Completeness	Adjusted Rand Index	Fowlkes-Mallows Score	Accuracy	Completeness	Adjusted Rand Index	Fowlkes-Mallows Score	Accuracy	Completeness	Adjusted Rand Index	Fowlkes-Mallows Score
15 Points, 3 Clusters, 2 Dim				30 Points, 3 Clusters, 2 Dim				45 Points, 3 Clusters, 2 Dim				
SIM	56.4±1.6	79.5±0.8	74.7±1.0	81.9±0.7	38.6±1.5	74.2±0.8	73.3±0.9	81.6±0.6	50.8±1.6	86.2±0.5	87.3±0.5	91.4±0.3
K-means	51.3±1.6	75.0±0.9	68.8±1.1	77.7±0.8	37.0±1.5	71.9±0.8	70.2±0.9	79.4±0.6	52.8±1.1	85.9±0.4	86.6±0.4	90.9±0.3
15 Points, 3 Clusters, 2 Dim				10 Points, 2 Clusters, 2 Dim				20 Points, 4 Clusters, 4 Dim				
QA	56.1±1.6	79.4±0.8	74.6±1.0	81.9±0.7	74.3±1.4	80.2±1.1	79.6±1.1	88.7±0.6	12.4±1.0	51.2±0.8	34.0±1.0	47.9±0.8
SIM	56.4±1.6	79.5±0.8	74.7±1.0	81.9±0.7	74.1±1.4	80.0±1.1	79.5±1.1	88.6±0.6	32.4±1.5	70.4±0.8	59.2±1.1	67.8±0.8
K-means	51.3±1.6	75.0±0.9	68.8±1.1	77.7±0.8	70.1±1.4	76.3±1.2	75.4±1.2	86.3±0.7	20.9±1.3	61.9±0.8	47.4±1.0	58.5±0.8
Exhaustive	56.4±1.6	79.5±0.8	74.7±1.0	81.9±0.7	74.3±1.4	80.2±1.1	79.6±1.1	88.7±0.6	-	-	-	-

TABLE 7.1: Synthetic data results for our approach (QA, SIM, exhaustive) and k-means. All numbers in % with standard error of the mean.

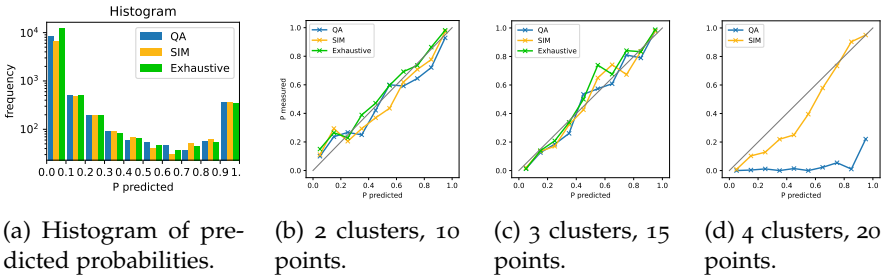


FIGURE 7.2: Evaluation of the calibration for QA, SIM and exhaustive search in a clustering scenario with 15 points and 3 clusters. Evaluation of the calibration for quantum annealing, simulated annealing and exhaustive search in clustering scenarios with 2, 3, and 4 clusters and 5 points in each cluster. All results are generated with 1,000 problems in each scenario and 5,000 measurements for each clustering problem.

7.4.3 Calibration Performance

We evaluate the calibration of our method on a synthetic clustering scenario with 3 clusters and 15 points using QA, SIM and exhaustive search on 1000 tasks. First all clustering solutions Z are accumulated in bins according to their estimated posterior probability $P(Z|X)$. This process also includes all sampled non-optimal but feasible solutions. The resulting histogram of solutions is shown in Figure 7.2a. After accumulation, the ratio of correct solutions in each bin is evaluated in Figures 7.2b, 7.2c, and 7.2d, which for a calibrated method should be close to the mean predicted probability represented by the diagonal. We find our approach to generate well-calibrated probabilities in both simulation and when using QA.

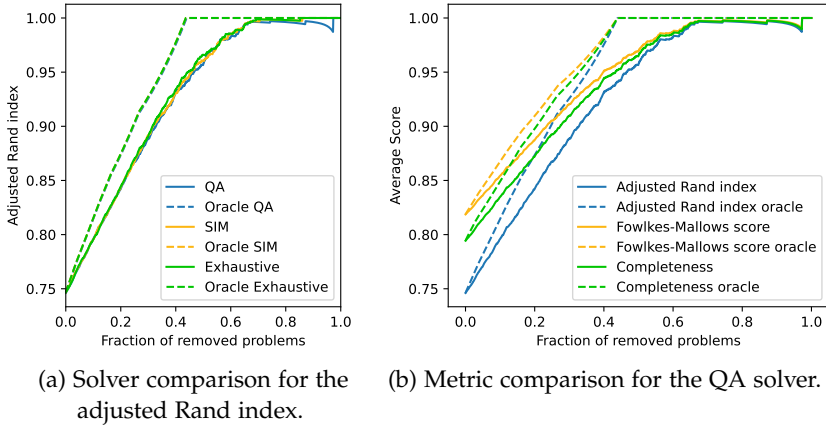


FIGURE 7.3: Sparsification plots of clustering metrics.

7.4.4 Clustering Performance

We evaluate the performance of our clustering formulation using synthetic data in Table 7.1. The upper rows show results for 3 clusters with an increasing number of points and 1000 tasks/setting for SIM and k-means. Due to the problem size, results cannot be found using QA and exhaustive search. Comparing results shows that most clustering metrics are better when using our formulation with SIM, however, the difference decreases with increasing problem size. This can be attributed to the globally optimal solution our approach is optimizing for. While in an ideal the global solution is at least as good as the k-means solution, an increasing problem size also increases the complexity of the corresponding QUBO. This makes it harder to solve exactly and the noisy approach using annealing is not capable of solving the problem correctly.

For the largest clustering problem with 3 clusters and 45 points, k-means provides the best accuracy and thus, a larger number of correct solutions compared to SIM. However, the metrics indicating clustering quality are higher for SIM. This indicates that our formulation is able to find a better solution in cases where the problem is not solved correctly by SIM and k-means.

The lower rows in Table 7.1 show settings with 2,3, and 4 clusters, each containing 5 points, again with 1000 tasks/setting. For the two smaller scenarios QA on the D-wave Advantage 2 provides results close or identical

to SIM and exhaustive search. For the largest scenario with 20 points in 4 clusters, QA loses performance.

The quality of predicted posterior probabilities can be further evaluated by linking them to the clustering metrics and sparsifying the set of tasks. Starting with metrics over the whole set of clustering tasks, we remove tasks according to increasing predicted probability and evaluate the metric over the remaining set. In an ideal predictor, this removes the lowest performing tasks first. In Figures 7.3a and 7.3b this is done for the adjusted Rand index with different solvers and for QA with different metrics respectively. The solid lines show the sparsification-plots using the predicted probabilities and the dashed lines show the same plots for an oracle method that generates the best possible ordering based on the metrics themselves.

In Figure 7.3a it becomes apparent that QA, SIM and exhaustive search perform close to each other over most of the value range. Nevertheless, for a high sparsification with more than 80% of the tasks removed, QA shows a drop in performance compared to the other methods. This is caused by tasks where only a single, but incorrect solution is found, which gets assigned a posterior probability of $P(Z|\mathbf{X}) = 1.0$. In such cases the Boltzmann distribution has not been sampled sufficiently well, either because of too few measurements or because of a low effective sampling temperature. As SIM does not show this behavior the source can likely be traced back to current limitation of the quantum computer.

7.4.5 *Maximum Pointsets*

Qualitative examples for the maximum pointsets generated with Algorithm 1 are depicted in Figure 7.4. The shape of each point represents the ground truth class and the color the assigned cluster. Starting with the most likely solution having a predicted probability of $p(Y|X) = 0.61$ on the left, each plot shows one additional step of the algorithm, which successively removes points from the solution, indicated by plotting them in Grey. The illustration demonstrates that the MaxsetSearch algorithm is able to generate well-separated clusters from the probabilistic predictions.

7.4.6 *IRIS Dataset*

While our method assumes an identity covariance in the data, it can be applied to other distributions, which we evaluate on the widely used IRIS dataset. Clustering metrics for experiments using 3 clusters with 5 points

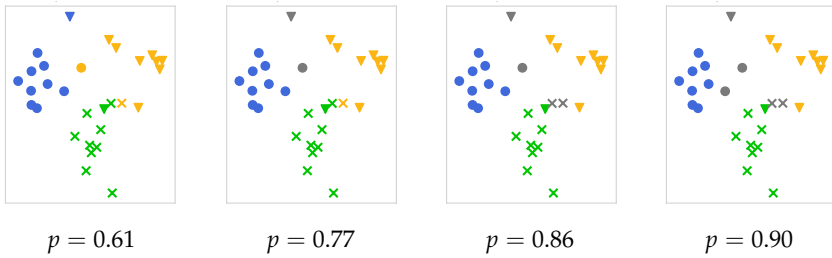
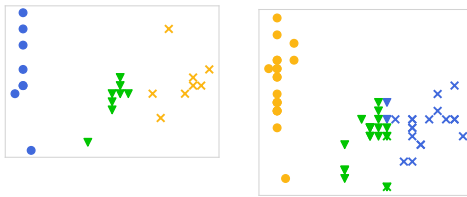


FIGURE 7.4: Visualization of max pointsets for synthetic data with the probability for each of the determined pointsets.



(a) 24 points, solved using QA. (b) 60 points, solved using SIM.

FIGURE 7.5: Qualitative results on the IRIS dataset.

	Accuracy	Completeness	Adjusted Rand Index	Fowlkes-Mallows Score
QA	47.2	81.7	76.8	83.4
SIM	47.1	81.7	76.7	83.4
K-means	47.0	80.6	75.5	82.5
Exhaustive	47.1	81.8	76.8	83.5

TABLE 7.2: Clustering metrics generated on IRIS subsets using 3 clusters with 15 points in total.

each are provided in Table 7.2 and show that our formulation using QA and SIM is competitive with k-means. Figures 7.5a and 7.5b show differently sized qualitative examples from the dataset solved using our formulation with QA and SIM respectively.

Though the value of the predicted probability cannot be interpreted easily with the mismatch between the assumed and the actual data-generating process, it can be used to find maximum pointsets. The results from Algorithm 1 using results from SIM are provided in Figure 7.6 and show that even for a distribution mismatch the maximum pointsets can provide meaningful results for removing ambiguous samples.

7.5 CONCLUSION

In this chapter, we described a probabilistic clustering approach based on sampling k-means solutions using AQC. By using all valid measurements,

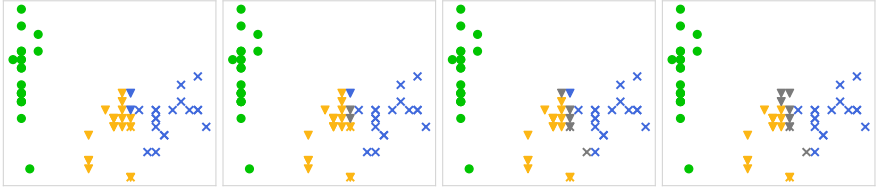


FIGURE 7.6: Max pointsets generated using simulated annealing on the IRIS dataset with 60 points.

calibrated confidence scores are computed at little cost and solutions are competitive to an iterative balanced k-means approach. We evaluated our method on synthetic as well as real data using simulation, exhaustive search as well as the D-Wave Advantage 2 prototype AQC to explore the potential of quantum computing in machine learning and computer vision.

CONCLUSION

8.1 SUMMARY OF CONTRIBUTIONS

In this thesis, computer vision for autonomous systems has been investigated on three levels of abstraction. In Part I high-level scene understanding was discussed with applications in autonomous driving and robot fleets.

In Chapter 2, a method to extract high-level driving actions from a dataset containing vehicle trajectories together with semantic map information for autonomous driving applications was proposed. Based on an HMM formulation of an agent's activity, we were able to decode the underlying sequence of states and thus, were able to understand the corresponding driving actions. As the state-decoding relies on the map information, manual work is only required for annotating the map, rather than each agent's trajectory, which makes the method scalable for large datasets. Using the extracted actions a prediction module that used the rendered agent trajectory together with map views has been trained. With the human-interpretable action classes being available, we were able to evaluate the prediction model together with the properties of the underlying data.

In Chapter 3, we developed a full action recognition pipeline that has been deployed on a humanoid autonomous robot. The major contributions of the work were 1) the collection of a multi-domain dataset for referee action recognition that is easily adaptable to changes in the environment and rules. It contains three distinct domains, from fully synthetic images that can easily be adapted to rule changes, to a fully manually annotated real domain that was used to test the method in a competition scenario. 2) The development of a highly efficient action recognition model and the deployment of it on the NAO robot. Embedded into the full autonomous software framework, we showed that it can perform real-time inference with a strongly limited compute budget. 3) An in-depth evaluation of the dataset design based on the developed model. It showed that using simulated and real data together yields the best model and that combining orthogonal domain gaps can provide a considerable performance improvement.

In Part II, we were diving deeper into tracking, one of the prerequisites for action detection and recognition in autonomous vehicles.

In Chapter 4, predictive models for the state of a track were combined with a graph NMP framework for object matching. This allowed us to extend a previous NMP approach [28] to online tracking of objects in 3D. The graph structure implemented two matching problems in parallel, the matching of detections across multiple frames as well as the matching of active tracks to detections. The active tracks are parameterized by a predictive model and thus, allow the interpolation of occlusions, as well as provide a strong signal for matching. By performing experiments on the nuScenes dataset [31], we showed that our approach considerably improves track stability and thus, the overall tracking performance. During the AI Driving Olympics Tracking Challenge at the IEEE International Conference on Robotics and Automation 2021, we were able to achieve the second place for 3D trackers and the best LIDAR-only method.

In Chapter 5, the challenge of long-term tracking was approached in the context of RoboCup. This was an application that came with the challenges also encountered in any real-world scenario where a fleet of autonomous robots has to be tracked from an uncalibrated camera. These included long and strong occlusions, identically looking robots with labels that are too small to read from an external view and intersecting robots that can easily induce ID-switches. To solve the task we built a tracking pipeline that combined information acquired on each robot with a deep-learning short-term tracker [18]. The fusion of multi-platform information has been achieved by a discrete optimization problem mapping tracklets to robot IDs. To solve the non-trivial and multi-dimensional cost weighting, we employed PSO to find the optimal weights. We evaluated the importance of the different features and found that redundant information is available in different signals, which can be used to make the method more robust. Finally, we matched the cost-weighting to different sequences, which allowed us to qualitatively understand the shortcomings of each team's algorithms in each of the videos.

Like most methods in computer vision, also our approaches come at a high computational cost, at least during model training. Thus, we investigated quantum computing in computer vision and machine learning in Part III of this thesis, which provides a way to unlock large amounts of computational power.

In Chapter 6, we applied this paradigm to MOT, where we derived a formulation of the tracking assignment problem suitable for an AQC. As the solution probability and thus, the number of measurements required to solve the optimization problem is strongly related to the spectral gap of

the cost function, we investigated the influence of the constraints on the cost matrix. Based on the results, we proposed an iterative optimization scheme as well as an initialization method for the constraint weighting. We demonstrated that our formulation is competitive on a common MOT benchmark [134] by using a classical solver, even though the task is much harder in this context. In simulation and on a D-Wave QA we analyzed the properties of the measurement results and demonstrated that small real-world examples of MOT can already be solved on a real quantum computer.

Finally, we approached the efficient use of an AQC for clustering tasks in Chapter 7. From the observation that previous approaches in quantum computer vision used the QA purely as an optimizer, we derived a quantum k-means clustering formulation that utilizes all measurements performed on an AQC during the optimization process. The derived approach provides a set of alternative solutions to the clustering task, together with calibrated confidence scores, while only incurring little additional compute cost. We furthermore proposed an algorithm that merges the set of found clustering solutions, to remove ambiguous points. Our experiments demonstrated the valid calibration of the approach in simulation and for small problem sizes also with a D-Wave QA. Finally, using the IRIS dataset, we showed that our algorithm was able to remove ambiguous points and can be applied to distributions that do not fully follow the initial assumption of the distribution.

8.2 DISCUSSION, LIMITATIONS AND FUTURE WORK

In this thesis, we proposed approaches for the computer vision pipeline of an autonomous system that is used to understand its surroundings and environment, with a special focus on MOT and prediction tasks. Furthermore, we investigated the efficient use of quantum computing for these applications. We believe that there are several ways to extend and improve our proposed approaches which are discussed in the following.

8.2.1 *Action Sequence Predictions of Vehicles in Urban Environments*

In this chapter we explored the task of predicting high-level actions of vehicles in urban environments. To make data annotation for this task tractable, we proposed an algorithm to automatically extract action sequences with a semantic map and used it to annotate the public Argoverse [39] dataset.

A fundamental design choice was the use of a CNN model with late fusion to perform action prediction. As the late-fusion approach uses a feature extractor on each frame, it allows us to easily adapt the architecture to new models proposed for feature extraction. Thus, one promising direction is the integration of a transformer backbone network like ViT [61]. Such a design change can be motivated from two perspectives. First, transformers have recently shown a considerable performance boost compared to CNNs and thus, the quality of extracted features can be considered to be better. Secondly, transformers are designed to learn the relation between different parts of the image. For a driving and prediction scenario, this is information that needs to be modeled i.e. the relation of the ego-vehicle to the road layout as well as to the other traffic participants strongly influences the probability of future actions, which would make ViT a well-grounded choice as a feature extractor.

While the selection of the architecture provides flexibility in adapting the model to new progress, it requires to be run for each agent separately to perform predictions. This is especially challenging with a computationally heavy feature extractor in dense traffic. One possible solution to this problem is the extraction of a global feature map for a large scene that contains all traffic participants, and the extraction of relevant local features from the global feature map during late fusion. This allows to only run the lightweight part of the model for each participant separately while sharing the backbone. One challenge, especially for CNNs, is the large receptive field that is required in this approach. However, the combination with transformers and thus an explicit model of relations could potentially alleviate this drawback.

Finally, high-level action predictions are well interpretable by humans and can be used well together with HD-map data, however, the combination with trajectory prediction in an autonomous driving framework provides great potential to get the best of both worlds. While the trajectories themselves are important for path planning, high-level actions can help to quantify the confidence of trajectories. This can support the sampling of diverse futures, which can enable explicitly risk-aware planning.

8.2.2 *Recognition of Referee Signals on Robotic Platforms*

Our approach towards recognizing referee actions on the humanoid NAO robot included the collection and annotation of a dataset as well as the development of a computationally efficient detection model on the robot.

The dataset contains four domains, from a fully simulated one to a real domain that resembles the scenarios also encountered during World Cup matches.

One important aspect to improve our approach is the estimation of a confidence score for the robot's prediction. The evaluation shows that there is a considerable performance difference between predictions from images that show a frontal and clear view of the referee and images that are taken under deteriorated conditions. These images are either taken from the side of the field, too close to the referee, or under bad lighting conditions, e.g. with strong illumination changes in the background. Knowing if a robot has been able to make a reliable prediction can be used as a factor to decide on the next step in the game strategy. We believe that this problem should be approached using two kinds of signals; First, with the position on the field from self-localization, which directly provides information about the quality of the viewing angle. Second, with the image itself as this is the only way to get information about the lighting conditions, or possible localization errors that move the referee out of the image.

Based on this, another promising direction to extend our action detection method is to fuse information from multiple robots that play in one team. While the model currently only considers a single robot for recognizing the referee's action, there are always multiple robots deployed during a match. Therefore, fusing the information acquired on each of the robots to make a global decision could improve the performance considerably. This could be studied under different constraints, from full Wi-Fi communication where rich information can be shared to limited protocols where only sparse messages can be transmitted between robots.

Finally, domain adaptation can be studied in our dataset. The combination of different domains was already investigated in the respective chapter and was primarily aimed at demonstrating how data collection and annotation cost can be reduced. However, another aspect is the adaptation to rule changes in RoboCup where e.g. new actions are introduced. In this case, the adaptation from a fully synthetic domain, which can easily be changed to reflect the new rules, to the real domain is crucial. An important aspect that needs to be considered for this scenario is that real data would only be available for a subset of the relevant actions.

8.2.3 *Learnable Online Graph Representations for 3D MOT*

For 3D MOT, we proposed an NMP-based tracking method that combined predictive models with fully learnable object matching. Possible improvements of this framework can be found by starting with typical failure cases that we observed. Typical scenarios where our tracker failed are as follows:

1. Long frame gaps cause scenarios where the time a track is kept active without observations is shorter than the observed occlusion time. While extending this timeframe may further reduce the number of ID-switches, it also increases the number of false positives, harming the overall tracking performance.
2. Consistent false positive detections are a general problem for any tracker following the tracking-by-detection paradigm. While our tracker can easily handle isolated false positives due to noise in the detector, cases where e.g. physical structures or reflections are detected cannot be recognized.
3. Double objects are generated if the same object is detected multiple times as different types. This behavior can mostly be observed for trucks in our results and should be approached at the detector level e.g. with non-maximum suppression [97].

It is important to note that all of these scenarios are also failure cases for existing trackers and are not newly introduced by our pipeline.

While the first type of error needs to be addressed primarily by improving evaluation metrics, which often do not cover occluded scenarios, it also provides valuable impulses for extending our tracker. Covering long frame gaps is especially important in autonomous driving, where knowledge about traffic participants that are currently unobserved, but can reappear is needed for planning. A possible solution to this is the use of an occupancy map that also represents unobserved areas and thus allows one to model long occlusions without adding additional false positives.

Closely related to this is the integration of a semantic map, which can be used to improve tracking in two aspects. First, a large fraction of encountered failure cases can be attributed to consistent false positives generated by the object detector. As these stem from static objects or reflections, they do not follow valid trajectories on a map and thus, can be eliminated. Secondly, map data can be used in the predictive model, which forms a core component of our approach, to provide improved trajectory estimates that follow the rules of the road.

8.2.4 *Long-Term Robot Tracking with Multi-Platform Sensor Fusion*

Following the goal of performing stable long-term tracking, this chapter presented a method for tracking multiple similar humanoid robots using sensor fusion. It utilized information from the robots' sensors together with an external camera view by combining them in a quadratic problem.

One open challenge in our approach is the adaptation of the cost weighting to the different algorithms running on the robots. As robots play for different teams, their respective methods are deployed during the games performing all relevant tasks and thus, different teams exhibit different performance levels on these. This includes self-localization and fall detection which are used as cost terms in our optimization framework. The weighting of the costs is currently optimized over the complete training set, however, given sufficient data, a possible extension can adapt them to the performance of the playing teams.

Another important aspect with a large potential for improvement is the detection of the IDs printed on the robots, as they uniquely identify them. While detecting the ID is an easy task in a high-resolution video, it becomes much harder to solve with a wide-angle camera, as used in our target application to cover a large space efficiently. A possible approach is based on the current improvements in video-superresolution. As the robots are moving, slightly different views are available in each frame. Together with short-term tracking, this can be used as the input to such a method and possibly provide images that are sufficient for detecting robot numbers. In this case, the advantage over directly training a number detector on a frame sequence is the utilization of existing superresolution-datasets without additional annotation effort.

8.2.5 *Adiabatic Quantum Computing for Multi-Object Tracking*

In this chapter, a quantum computing formulation of MOT suitable for an AQC has been proposed. It was based on stating the matching problem as QUBO and further investigated suitable formulations of the clustering constraints.

Some limitations of the proposed formulation arise from it being optimized to run on current AQCs. As QUBO is a hard problem to solve with classical approaches and as current AQCs are still at a small scale, problem sizes are also limited. Nevertheless, the increasing size of quantum computers promises to make much larger problems feasible in the future.

One important extension of the algorithm is the introduction of sparsity in the Hamiltonian function. In our current formulation, a dense form is used as it provides the scope to encode all information available into a single representation. However, as the qubits on an AQC are only locally connected to their neighbors, the scaling of this approach to larger problems is limited by the required mapping function. Sparse representations that solve this problem could use information only about pairs of objects that are strongly similar or strongly different. However, they also need to consider the encoding of constraints, as these also require dense connections in their current form.

Another aspect aimed at increasing the feasible problem size is the development of an iterative solver that first solves for easy to generate tracks, and uses the AQC to solve the optimization problem of hard decisions. Such an approach has been considered in Q-match [16], where it increased the scale of feasible problems by one magnitude and could also be transferred to a multi-object tracking formulation similar to ours.

8.2.6 Probabilistic Sampling of Balanced K-Means using AQC

In the last chapter, we described a method to perform sampling of probabilistic k-means on an AQC. Using our formulation, we were able to computer calibrate confidence scores by using all valid measurements on the AQC, which were unused and discarded in previous approaches.

Similar to the previous chapter, sparse formulations of the task should be investigated are they are important to allow the problem size to scale up further. In this case, a special focus needs to be put on not removing any relations that are decisive for switching cluster assignments.

While our method and experiments were aimed at k-means clustering as a fundamental machine-learning problem and the basis of unsupervised algorithms in computer vision, the formulation can readily be transferred to other applications in computer vision. E.g. our MOT approach as well as Birdal *et al.* [23] and Benkner *et al.* [17] can be cast to a balanced clustering problem with additional constraints. Implementing these constraints can be performed by adapting the prior distribution that influences the energy formulation, however, more work would be needed to achieve a sufficiently high solution probability on the current noisy AQCs.

Further approaching the constraints, our formulation can be extended to non-balanced k-means clustering. Similarly to the implementation of problem-specific constraints, this would be approached through the adap-

tion of the prior distribution. In this case, additional states need to be allowed which increases the scope of feasible solutions and thus, likely also the number of measurements required to sample the distribution sufficiently well.

Following the k-means formulation and thus, assuming Gaussian clusters is a well-justified choice and results in the use of the quadratic Euclidean distance for the energy function. Nevertheless, in the QUBO formulation, using a quadratic distance metric introduces large differences in the components of the cost matrix. As the cost matrix needs to be scaled to fit the range that can be represented by the Ising model implemented on the AQC, smaller components are compressed. However, these terms are the most relevant ones as they are included in the most likely solutions that we aim to sample. An important aspect to study is thus the extension of our algorithm to distance measures that better represent the connection of similar points and reduce the influence of outliers.

Overall, while the approach is still limited in the problem size, quantum computing enables a fundamentally different approach to clustering that can provide additional information that is costly to compute otherwise. With the current progress and potential to scale to real-world problems, more work is required to adapt existing problem formulations, such that the full capability of quantum computing can be used efficiently.

BIBLIOGRAPHY

1. Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A. & Woerner, S. The Power of Quantum Neural Networks. *Nature Computational Science* **1**, 403 (2021).
2. Abdel-Aziz, Y. & Karara, H. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry. *Photogrammetric Engineering & Remote Sensing* **81**, 103 (2015).
3. Agarap, A. F. Deep Learning Using Rectified Linear Units (ReLU). *ArXiv* **abs/1803.08375** (2018).
4. Agarwal, P., Paudel, D. P., Zaech, J.-N. & Van Gool, L. *Unsupervised Robust Domain Adaptation Without Source Data* in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), 2009.
5. Aïmeur, E., Brassard, G. & Gambs, S. *Quantum Clustering Algorithms* in *Proceedings of the 24th International Conference on Machine Learning - ICML '07* (ACM Press, Corvallis, Oregon, 2007), 1.
6. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L. & Savarese, S. *Social LSTM: Human Trajectory Prediction in Crowded Spaces* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 961.
7. Alberts, G. J. N., Rol, M. A., Last, T., Broer, B. W., Bultink, C. C., Rijlaarsdam, M. S. C. & Van Hauwermeiren, A. E. Accelerating Quantum Computer Developments. *EPJ Quantum Technology* **8**, 1 (2021).
8. Alvarez, L., Gómez, L. & Sendra, J. R. An Algebraic Approach to Lens Distortion by Line Rectification. *Journal of Mathematical Imaging and Vision* **35**, 36 (2009).
9. Apolloni, B., Carvalho, C. & de Falco, D. Quantum Stochastic Optimization. *Stochastic Processes and their Applications* **33**, 233 (1989).
10. Arrigoni, F., Menapace, W., Benkner, M. S., Ricci, E. & Golyanik, V. in *Computer Vision – ECCV 2022* (eds Avidan, S., Brostow, G., Cissé, M., Farinella, G. M. & Hassner, T.) 506 (Springer Nature Switzerland, Cham, 2022).

11. Arthur, D. & Date, P. Balanced K-Means Clustering on an Adiabatic Quantum Computer. *Quantum Information Processing* **20**, 294 (2021).
12. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M. P., Hartmann, M. J., Ho, A., Hoffmann, M., Huang, T., Humble, T. S., Isakov, S. V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P. V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M. Y., Ostby, E., Petukhov, A., Platt, J. C., Quintana, C., Rieffel, E. G., Roushan, P., Rubin, N. C., Sank, D., Satzinger, K. J., Smelyanskiy, V., Sung, K. J., Trevithick, M. D., Vainsencher, A., Villalonga, B., White, T., Yao, Z. J., Yeh, P., Zalcman, A., Neven, H. & Martinis, J. M. Quantum Supremacy Using a Programmable Superconducting Processor. *Nature* **574**, 505 (2019).
13. Bak, S., Carr, P. & Lalonde, J.-F. *Domain Adaptation through Synthesis for Unsupervised Person Re-Identification in Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII* (eds Ferrari, V., Hebert, M., Sminchisescu, C. & Weiss, Y.) **11217** (Springer, 2018), 193.
14. Bansal, M., Krizhevsky, A. & Ogale, A. *ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst in Robotics: Science and Systems XV* (Robotics: Science and Systems Foundation, 2019).
15. Bar-Shalom, Y., Daum, F. & Huang, J. The Probabilistic Data Association Filter. *IEEE Control Systems Magazine* **29**, 82 (2009).
16. Benkner, M., Lahner, Z., Golyanik, V., Wunderlich, C., Theobalt, C. & Moeller, M. *Q-Match: Iterative Shape Matching via Quantum Annealing in 2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE Computer Society, Los Alamitos, CA, USA, 2021), 7566.
17. Benkner, M. S., Golyanik, V., Theobalt, C. & Moeller, M. *Adiabatic Quantum Graph Matching with Permutation Matrix Constraints in 2020 International Conference on 3D Vision (3DV)* (IEEE, Fukuoka, Japan, 2020), 583.

18. Bergmann, P., Meinhardt, T. & Leal-Taixe, L. *Tracking Without Bells and Whistles* in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, Seoul, Korea (South), 2019), 941.
19. Bermejo, P. & Orús, R. Variational Quantum and Quantum-Inspired Clustering. *Scientific Reports* **13**, 13284 (2023).
20. Bewley, A., Ge, Z., Ott, L., Ramos, F. & Upcroft, B. *Simple Online and Realtime Tracking* in *2016 IEEE International Conference on Image Processing (ICIP)* (2016), 3464.
21. Bhatia, H., Tretschk, E., Lähner, Z., Benkner, M. S., Moeller, M., Theobalt, C. & Golyanik, V. *CCuantuMM: Cycle-Consistent Quantum-Hybrid Matching of Multiple Shapes* in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Vancouver, BC, Canada, 2023), 1296.
22. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. & Lloyd, S. Quantum Machine Learning. *Nature* **549**, 195 (2017).
23. Birdal, T., Golyanik, V., Theobalt, C. & Guibas, L. *Quantum Permutation Synchronization* in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Nashville, TN, USA, 2021), 13117.
24. Bobick, A. F. & Davis, J. W. The Recognition of Human Movement Using Temporal Templates. *IEEE Transactions on pattern analysis and machine intelligence* **23**, 257 (2001).
25. Bolelli, F., Allegretti, S., Baraldi, L. & Grana, C. Spaghetti Labeling: Directed Acyclic Graphs for Block-Based Connected Components Labeling. *IEEE Transactions on Image Processing* **29**, 1999 (2020).
26. Born, M. & Fock, V. Beweis des Adiabatenatzes. *Zeitschrift für Physik* **51**, 165 (1928).
27. Boroushaki, T., Perper, I., Nachin, M., Rodriguez, A. & Adib, F. *RFusion: Robotic Grasping via RF-Visual Sensing and Learning* in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems* (Association for Computing Machinery, New York, NY, USA, 2021), 192.
28. Braso, G. & Leal-Taixe, L. *Learning a Neural Solver for Multiple Object Tracking* in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Seattle, WA, USA, 2020), 6246.

29. Bunyk, P. I., Hoskinson, E., Johnson, M. W., Tolkacheva, E., Altomare, F., Berkley, A. J., Harris, R., Hilton, J. P., Lanting, T. & Whittaker, J. Architectural Considerations in the Design of a Superconducting Quantum Annealing Processor. *IEEE Transactions on Applied Superconductivity* **24**, 1 (2014).
30. Byrd, G. T. & Ding, Y. Quantum Computing: Progress and Innovation. *Computer* **56**, 20 (2023).
31. Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G. & Beijbom, O. *nuScenes: A Multimodal Dataset for Autonomous Driving in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020* (IEEE, 2020), 11618.
32. Campos, C., Elvira, R., Rodríguez, J. J. G., M. Montiel, J. M. & D. Tardós, J. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics* **37**, 1874 (2021).
33. Caron, M., Bojanowski, P., Joulin, A. & Douze, M. in *Computer Vision – ECCV 2018* 139 (Springer International Publishing, Cham, 2018).
34. Carreira, J. & Zisserman, A. *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 6299.
35. Caruana, R., Lawrence, S. & Giles, L. *Overfitting in Neural Nets: Back-propagation, Conjugate Gradient, and Early Stopping in Proceedings of the 13th International Conference on Neural Information Processing Systems* (MIT Press, Cambridge, MA, USA, 2000), 381.
36. Casaña-Eslava, R. V., Lisboa, P. J. G., Ortega-Martorell, S., Jarman, I. H. & Martín-Guerrero, J. D. *A Probabilistic Framework for Quantum Clustering* 2019.
37. Casaña-Eslava, R. V., Lisboa, P. J. G., Ortega-Martorell, S., Jarman, I. H. & Martín-Guerrero, J. D. Probabilistic Quantum Clustering. *Knowledge-Based Systems* **194**, 105567 (2020).
38. Casas, S., Luo, W. & Urtasun, R. *IntentNet: Learning to Predict Intention from Raw Sensor Data in Proceedings of The 2nd Conference on Robot Learning* (PMLR, 2018), 947.

39. Chang, M.-F., Ramanan, D., Hays, J., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P. & Lucey, S. *Argoverse: 3D Tracking and Forecasting With Rich Maps in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Long Beach, CA, USA, 2019), 8740.
40. Chi, Y., Huang, J., Zhang, Z., Mao, J., Zhou, Z., Chen, X., Zhai, C., Bao, J., Dai, T., Yuan, H., Zhang, M., Dai, D., Tang, B., Yang, Y., Li, Z., Ding, Y., Oxenløwe, L. K., Thompson, M. G., O'Brien, J. L., Li, Y., Gong, Q. & Wang, J. A Programmable Qudit-Based Quantum Processor. *Nature Communications* **13**, 1166 (2022).
41. Chin, T.-J., Suter, D., Ch'ng, S.-F. & Quach, J. in *Computer Vision – ACCV 2020* (eds Ishikawa, H., Liu, C.-L., Pajdla, T. & Shi, J.) 485 (Springer International Publishing, Cham, 2021).
42. Chiu, H.-k., Li, J., Ambrus, R. & Bohg, J. Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving. *IEEE International Conference on Robotics and Automation (ICRA)* (2021).
43. Chiu, H.-k., Prioletti, A., Li, J. & Bohg, J. Probabilistic 3D Multi-Object Tracking for Autonomous Driving. *arXiv:2001.05673 [cs]* (2020).
44. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. *Empirical evaluation of gated recurrent neural networks on sequence modeling in NIPS 2014 Workshop on Deep Learning, December 2014* (2014).
45. Coates, A. & Ng, A. Y. in *Neural Networks: Tricks of the Trade* (eds Montavon, G., Orr, G. B. & Müller, K.-R.) 561 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).
46. Coleman, G. & Andrews, H. Image Segmentation by Clustering. *Proceedings of the IEEE* **67**, 773 (1979).
47. D-Wave Systems Inc. *Performance Advantage in Quantum Boltzmann Sampling* tech. rep. (D-Wave Systems Inc., 2017).
48. Dai, P., Weng, R., Choi, W., Zhang, C., He, Z. & Ding, W. *Learning a Proposal Classifier for Multiple Object Tracking in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Nashville, TN, USA, 2021), 2443.
49. Das, S., Wildridge, A. J., Vaidya, S. B. & Jung, A. Track Clustering with a Quantum Annealer for Primary Vertex Reconstruction at Hadron Colliders. *arXiv:1903.08879 [hep-ex, physics:quant-ph]* (2020).
50. Date, P., Arthur, D. & Pusey-Nazzaro, L. QUBO Formulations for Training Machine Learning Models. *Scientific Reports* **11**, 10029 (2021).

51. De Souza, C. R., Gaidon, A., Cabon, Y. & Lopez, A. M. *Procedural Generation of Videos to Train Deep Action Recognition Networks* in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE, Honolulu, HI, 2017), 2594.
52. Debnath, S., Linke, N. M., Figgatt, C., Landsman, K. A., Wright, K. & Monroe, C. Demonstration of a Small Programmable Quantum Computer with Atomic Qubits. *Nature* **536**, 63 (2016).
53. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K. & Leal-Taixé, L. MOT20: A Benchmark for Multi Object Tracking in Crowded Scenes. *arXiv:2003.09003 [cs]* (2020).
54. Deo, N., Rangesh, A. & Trivedi, M. M. How Would Surround Vehicles Move? A Unified Framework for Maneuver Classification and Motion Prediction. *IEEE Transactions on Intelligent Vehicles* **3**, 129 (2018).
55. Deo, N. & Trivedi, M. M. *Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver Based LSTMs* in IEEE Intelligent Vehicles Symposium (IV) (2018).
56. Dhanachandra, N., Manglem, K. & Chanu, Y. J. Image Segmentation Using K -Means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Computer Science. Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India* **54**, 764 (2015).
57. Diamond, S. & Boyd, S. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research* **17**, 2909 (2016).
58. Dixit, V., Selvarajan, R., Alam, M. A., Humble, T. S. & Kais, S. Training Restricted Boltzmann Machines With a D-Wave Quantum Annealer. *Frontiers in Physics* **9** (2021).
59. Doan, A.-D., Sasdelli, M., Suter, D. & Chin, T.-J. *A Hybrid Quantum-Classical Algorithm for Robust Fitting* in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE, New Orleans, LA, USA, 2022), 417.

60. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. & Darrell, T. *Long-Term Recurrent Convolutional Networks for Visual Recognition and Description in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), 2625.
61. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. *An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale in International Conference on Learning Representations* (2021).
62. Efros, Berg, Mori & Malik. *Recognizing Action at a Distance in Proceedings Ninth IEEE International Conference on Computer Vision* (IEEE, 2003), 726.
63. Einstein, A., Podolsky, B. & Rosen, N. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Physical Review* **47**, 777 (1935).
64. Engel, J., Schöps, T. & Cremers, D. *LSD-SLAM: Large-scale Direct Monocular SLAM in Computer Vision – ECCV 2014* (eds Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T.) (Springer International Publishing, Cham, 2014), 834.
65. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (AAAI Press, Portland, Oregon, 1996), 226.
66. Farina, M., Magri, L., Menapace, W., Ricci, E., Golyanik, V. & Arigoni, F. *Quantum Multi-Model Fitting in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Vancouver, BC, Canada, 2023), 13640.
67. Feld, S., Friedrich, M. & Linnhoff-Popien, C. *Optimizing Geometry Compression Using Quantum Annealing in 2018 IEEE Globecom Workshops (GC Wkshps)* (2018), 1.
68. Feynman, R. P. Simulating Physics with Computers. *International Journal of Theoretical Physics* **21**, 467 (1982).

69. Fisher, J., Christen, P., Wang, Q. & Rahm, E. *A Clustering-Based Framework to Control Block Sizes for Entity Resolution in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, NY, USA, 2015), 279.
70. Fisher, R. A. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* **7**, 179 (1936).
71. Forgy, E. Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. *Biometrics* **21**, 768 (1965).
72. Fowlkes, E. B. & Mallows, C. L. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association* **78**, 553 (1983).
73. Fox, D. M., Branson, K. M. & Walker, R. C. mRNA Codon Optimization with Quantum Computers. *PLOS ONE* **16**, e0259101 (2021).
74. Gaidon, A., Wang, Q., Cabon, Y. & Vig, E. *Virtual Worlds as Proxy for Multi-Object Tracking Analysis in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 4340.
75. Ganian, R., Hamm, T. & Ordyniak, S. The Complexity of Object Association in Multiple Object Tracking. *Proceedings of the AAAI Conference on Artificial Intelligence* **35**, 1388 (2021).
76. Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research* **32**, 1231 (2013).
77. Gerke, S., Muller, K. & Schafer, R. *Soccer Jersey Number Recognition Using Convolutional Neural Networks in Proceedings of the IEEE International Conference on Computer Vision Workshops* (2015), 17.
78. Girdhar, R. & Ramanan, D. *Attentional Pooling for Action Recognition in Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., Red Hook, NY, USA, 2017), 33.
79. Granstrom, K. & Orguner, U. A Phd Filter for Tracking Multiple Extended Targets Using Random Matrices. *IEEE Transactions on Signal Processing* **60**, 5657 (2012).
80. Griffiths, D. J. & Schroeter, D. F. *Introduction to Quantum Mechanics* 3rd ed. (Cambridge University Press, Cambridge, 2018).

81. Grover, L. K. *A Fast Quantum Mechanical Algorithm for Database Search in Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, USA, 1996), 212.
82. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S. & Alahi, A. *Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Salt Lake City, UT, 2018), 2255.
83. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual 2021*.
84. Hahner, M., Dai, D., Sakaridis, C., Zaech, J.-N. & Gool, L. V. *Semantic Understanding of Foggy Scenes with Purely Synthetic Data in 2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019), 3675.
85. Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S. & Cipolla, R. *Understanding RealWorld Indoor Scenes with Synthetic Data in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Las Vegas, NV, USA, 2016), 4077.
86. Harris, R., Lanting, T., Berkley, A. J., Johansson, J., Johnson, M. W., Bunyk, P., Ladizinsky, E., Ladizinsky, N., Oh, T. & Han, S. Compound Josephson-junction Coupler for Flux Qubits with Minimal Crosstalk. *Physical Review B* **80**, 052506 (2009).
87. He, K., Gkioxari, G., Dollár, P. & Girshick, R. *Mask R-CNN in 2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 2980.
88. Hecker, S., Dai, D. & Gool, L. V. *End-to-End Learning of Driving Models with Surround-View Cameras and Route Planners in Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII* (eds Ferrari, V., Hebert, M., Sminchisescu, C. & Weiss, Y.) **11211** (Springer, 2018), 449.
89. Hensen, B., Bernien, H., Dréau, A. E., Reiserer, A., Kalb, N., Blok, M. S., Ruitenberg, J., Vermeulen, R. F. L., Schouten, R. N., Abellán, C., Amaya, W., Pruneri, V., Mitchell, M. W., Markham, M., Twitchen, D. J., Elkouss, D., Wehner, S., Taminiau, T. H. & Hanson, R. Loophole-Free Bell Inequality Violation Using Electron Spins Separated by 1.3 Kilometres. *Nature* **526**, 682 (2015).
90. Hirzer, M., Beleznai, C., Roth, P. M. & Bischof, H. in *Image Analysis* (eds Heyden, A. & Kahl, F.) 91 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011).

91. Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. & Darrell, T. *CyCADA: Cycle-Consistent Adversarial Domain Adaptation in Proceedings of the 35th International Conference on Machine Learning* (PMLR, 2018), 1989.
92. Hong, J., Sapp, B. & Philbin, J. *Rules of the Road: Predicting Driving Behavior with a Convolutional Model of Semantic Interactions in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
93. Hong, Y.-W. P., Huang, L.-M. & Li, H.-T. Vector Quantization and Clustered Key Mapping for Channel-Based Secret Key Generation. *IEEE Transactions on Information Forensics and Security* **12**, 1170 (2017).
94. Horn, D. & Gottlieb, A. *The Method of Quantum Clustering in Advances in Neural Information Processing Systems* (eds Dietterich, T., Becker, S. & Ghahramani, Z.) **14** (MIT Press, 2001).
95. Hornakova, A., Henschel, R., Rosenhahn, B. & Swoboda, P. *Lifted Disjoint Paths with Application in Multiple Object Tracking in Proceedings of the 37th International Conference on Machine Learning* (PMLR, 2020), 4364.
96. Hornakova, A., Kaiser, T., Swoboda, P., Rolinek, M., Rosenhahn, B. & Henschel, R. *Making Higher Order MOT Scalable: An Efficient Approximate Solver for Lifted Disjoint Paths in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 11.
97. Hosang, J., Benenson, R. & Schiele, B. *Learning Non-Maximum Suppression in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 6469.
98. Hossain, M. A., Pota, H. R., Squartini, S. & Abdou, A. F. Modified PSO Algorithm for Real-Time Energy Management in Grid-Connected Microgrids. *Renewable Energy* **136**, 746 (2019).
99. Houenou, A., Bonnifait, P., Cherfaoui, V. & Wen Yao. *Vehicle Trajectory Prediction Based on Motion Model and Maneuver Recognition in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2013).
100. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv abs/1704.04861* (2017).

101. Hoyer, L., Dai, D. & Van Gool, L. *DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation* in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New Orleans, LA, USA, 2022), 9914.
102. Hubert, L. & Arabie, P. Comparing Partitions. *Journal of Classification* **2**, 193 (1985).
103. IBM Quantum <https://quantum-computing.ibm.com/>. 2023.
104. Ising, E. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik* **31**, 253 (1925).
105. Jain, A., Koppula, H. S., Raghavan, B., Soh, S. & Saxena, A. *Car That Knows Before You Do: Anticipating Maneuvers via Learning Temporal Driving Models* in *IEEE International Conference on Computer Vision (ICCV)* (2015).
106. Jana, G., Mitra, A., Pan, S., Sural, S. & Chattaraj, P. K. Modified Particle Swarm Optimization Algorithms for the Generation of Stable Structures of Carbon Clusters. *Frontiers in Chemistry* **7** (2019).
107. Jhuang, H., Gall, J., Zuffi, S., Schmid, C. & Black, M. J. *Towards Understanding Action Recognition* in *Proceedings of the IEEE International Conference on Computer Vision* (2013), 3192.
108. Jiang, C., Cornman, A., Park, C., Sapp, B., Zhou, Y. & Anguelov, D. *MotionDiffuser: Controllable Multi-Agent Motion Prediction Using Diffusion* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 9644.
109. Jiang, N. & Wang, L. Quantum Image Scaling Using Nearest Neighbor Interpolation. *Quantum Information Processing* **14**, 1559 (2015).
110. Jozsa, R. & Linden, N. On the Role of Entanglement in Quantum-Computational Speed-Up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **459**, 2011 (2003).
111. Kadowaki, T. & Nishimori, H. Quantum Annealing in the Transverse Ising Model. *Physical Review E* **58**, 5355 (1998).
112. Kak, S. C. in *Advances in Imaging and Electron Physics* (ed Hawkes, P. W.) 259 (Elsevier, 1995).
113. Kalman, R. E. *et al.* A New Approach to Linear Filtering and Prediction Problems. *Journal of basic Engineering* **82**, 35 (1960).

114. Karthik, S., Prabhu, A. & Gandhi, V. Simple Unsupervised Multi-Object Tracking. *arXiv:2006.02609 [cs]* (2020).
115. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., *et al.* The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950* (2017).
116. Kesten, R., Usman, M., Houston, J., Pandya, T., Nadhamuni, K., Ferreira, A., Yuan, M., Low, B., Jain, A., Ondruska, P., Omari, S., Shah, S., Kulkarni, A., Kazakova, A., Tao, C., Platinsky, L., Jiang, W. & Shet, V. *Lyft Level 5 AV Dataset 2019* 2019.
117. Keuper, M., Tang, S., Andres, B., Brox, T. & Schiele, B. Motion Segmentation & Multiple Object Tracking by Correlation Co-Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**, 140 (2020).
118. Khosroshahi, A., Ohn-Bar, E. & Trivedi, M. M. *Surround Vehicles Trajectory Analysis with Recurrent Neural Networks in International Conference on Intelligent Transportation Systems (ITSC)* (2016).
119. Kim, A., Os̃ep, A. & Leal-Taixe, L. *EagerMOT: Real-time 3D Multi-Object Tracking and Segmentation via Sensor Fusion in IEEE International Conference on Robotics and Automation (ICRA)* (2021), 4.
120. Kim, B., Kang, C. M., Kim, J., Lee, S. H., Chung, C. C. & Choi, J. W. *Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (IEEE Press, Yokohama, Japan, 2017), 399.
121. King, A. D., Carrasquilla, J., Raymond, J., Ozfidan, I., Andriyash, E., Berkley, A., Reis, M., Lanting, T., Harris, R., Altomare, F., Boothby, K., Bunyk, P. I., Enderud, C., Fr chette, A., Hoskinson, E., Ladizinsky, N., Oh, T., Poulin-Lamarre, G., Rich, C., Sato, Y., Smirnov, A. Y., Swenson, L. J., Volkmann, M. H., Whittaker, J., Yao, J., Ladizinsky, E., Johnson, M. W., Hilton, J. & Amin, M. H. Observation of Topological Phenomena in a Programmable Lattice of 1,800 Qubits. *Nature* **560**, 456 (2018).
122. King, A. D., Raymond, J., Lanting, T., Isakov, S. V., Mohseni, M., Poulin-Lamarre, G., Ejtemaee, S., Bernoudy, W., Ozfidan, I., Smirnov, A. Y., Reis, M., Altomare, F., Babcock, M., Baron, C., Berkley, A. J., Boothby, K., Bunyk, P. I., Christiani, H., Enderud, C., Evert, B., Harris,

- R., Hoskinson, E., Huang, S., Jooya, K., Khodabandelou, A., Ladizinsky, N., Li, R., Lott, P. A., MacDonald, A. J. R., Marsden, D., Marsden, G., Medina, T., Molavi, R., Neufeld, R., Norouzpour, M., Oh, T., Pavlov, I., Perminov, I., Prescott, T., Rich, C., Sato, Y., Sheldan, B., Sterling, G., Swenson, L. J., Tsai, N., Volkmann, M. H., Whittaker, J. D., Wilkinson, W., Yao, J., Neven, H., Hilton, J. P., Ladizinsky, E., Johnson, M. W. & Amin, M. H. Scaling Advantage over Path-Integral Monte Carlo in Quantum Simulation of Geometrically Frustrated Magnets. *Nature Communications* **12**, 1113 (2021).
123. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* in *3rd International Conference on Learning Representations* (eds Bengio, Y. & LeCun, Y.) (2015).
124. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. Optimization by Simulated Annealing. *Science (New York, N.Y.)* **220**, 671 (1983).
125. Kitano, H. & Asada, M. The RoboCup Humanoid Challenge as the Millennium Challenge for Advanced Robotics. *Advanced Robotics* **13**, 723 (1998).
126. Kneip, L., Siegwart, R. & Scaramuzza, D. *A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation* in *2013 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, Los Alamitos, CA, USA, 2011), 2969.
127. Kocher, C. A. & Commins, E. D. Polarization Correlation of Photons Emitted in an Atomic Cascade. *Physical Review Letters* **18**, 575 (1967).
128. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T. & Serre, T. *HMDB: A Large Video Database for Human Motion Recognition* in *2011 International Conference on Computer Vision* (IEEE, 2011), 2556.
129. Kuete Meli, N., Mannel, F. & Lellmann, J. A Universal Quantum Algorithm for Weighted Maximum Cut and Ising Problems. *Quantum Information Processing* **22**, 279 (2023).
130. Laugier, C., Paromtchik, I. E., Perrollaz, M., Yong, M., Yoder, J.-D., Tay, C., Mekhnacha, K. & Nègre, A. Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety. *IEEE Intelligent Transportation Systems Magazine* **3**, 4 (2011).

131. Lazebnik, S., Schmid, C. & Ponce, J. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories* in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) **2** (2006), 2169.
132. Le, P. Q., Dong, F. & Hirota, K. A Flexible Representation of Quantum Images for Polynomial Preparation, Image Compression, and Processing Operations. *Quantum Information Processing* **10**, 63 (2011).
133. Le, P. Q., Iliyasu, A. M., Dong, F. & Hirota, K. Strategies for Designing Geometric Transformations on Quantum Images. *Theoretical Computer Science. Theoretical Computer Science Issues in Image Analysis and Processing* **412**, 1406 (2011).
134. Leal-Taixé, L., Milan, A., Reid, I., Roth, S. & Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv:1504.01942 [cs]* (2015).
135. Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H. S. & Chandraker, M. *DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, 2017), 2165.
136. Li, H.-S., Chen, X., Xia, H., Liang, Y. & Zhou, Z. A Quantum Image Representation Based on Bitplanes. *IEEE Access* **6**, 62396 (2018).
137. Li, H.-S., Fan, P., Xia, H.-Y., Peng, H. & Song, S. Quantum Implementation Circuits of Quantum Signal Representation and Type Conversion. *IEEE Transactions on Circuits and Systems I: Regular Papers* **66**, 341 (2019).
138. Li, J. & Ghosh, S. *Quantum-Soft QUBO Suppression for Accurate Object Detection* in *Computer Vision – ECCV 2020* (eds Vedaldi, A., Bischof, H., Brox, T. & Frahm, J.-M.) (Springer International Publishing, Cham, 2020), 158.
139. Li, X., Ying, X. & Chuah, M. C. *GRIP: Graph-based Interaction-aware Trajectory Prediction* in *IEEE Intelligent Transportation Systems Conference (ITSC)* (2019), 3960.
140. Li, Z., Yuan, L. & Nevatia, R. *Global Data Association for Multi-Object Tracking Using Network Flows* in 2008 IEEE Conference on Computer Vision and Pattern Recognition (2008), 1.

141. Liao, S., Hu, Y., Xiangyu Zhu & Li, S. Z. *Person Re-Identification by Local Maximal Occurrence Representation and Metric Learning* in 2015 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Boston, MA, USA, 2015), 2197.
142. Liao, Y., Qi, H. & Li, W. Load-Balanced Clustering Algorithm With Distributed Self-Organization for Wireless Sensor Networks. *IEEE Sensors Journal* **13**, 1498 (2013).
143. Lin, T., Goyal, P., Girshick, R., He, K. & Dollár, P. *Focal Loss for Dense Object Detection* in 2017 *IEEE International Conference on Computer Vision (ICCV)* (2017), 2999.
144. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. *Microsoft COCO: Common Objects in Context* in *Computer Vision – ECCV 2014* (eds Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T.) (Springer International Publishing, Cham, 2014), 740.
145. Lisnichenko, M. & Protasov, S. Quantum Image Representation: A Review. *Quantum Machine Intelligence* **5**, 2 (2022).
146. Liu, H. & Bhanu, B. *Pose-Guided R-CNN for Jersey Number Recognition in Sports* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2019).
147. Liu, H., Adreon, C., Wagnon, N., Bamba, A. L., Li, X., Liu, H., MacCall, S. & Gan, Y. Automated Player Identification and Indexing Using Two-Stage Deep Learning Network. *Scientific Reports* **13**, 10036 (2023).
148. Lloyd, S. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory* **28**, 129 (1982).
149. Lu, W.-L., Ting, J.-A., Little, J. J. & Murphy, K. P. Learning to Track and Identify Players from Broadcast Sports Videos. *IEEE transactions on pattern analysis and machine intelligence* **35**, 1704 (2013).
150. Ma, L., Tang, S., Black, M. J. & Van Gool, L. in *Computer Vision – ACCV 2018* (eds Jawahar, C. V., Li, H., Mori, G. & Schindler, K.) 612 (Springer International Publishing, Cham, 2019).
151. Ma, X.-S., Herbst, T., Scheidl, T., Wang, D., Kropatschek, S., Naylor, W., Wittmann, B., Mech, A., Kofler, J., Anisimova, E., Makarov, V., Jennewein, T., Ursin, R. & Zeilinger, A. Quantum Teleportation over 143 Kilometres Using Active Feed-Forward. *Nature* **489**, 269 (2012).

152. Maglo, A., Orcesi, A. & Pham, Q.-C. *Efficient Tracking of Team Sport Players with Few Game-Specific Annotations* in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (IEEE, New Orleans, LA, USA, 2022), 3460.
153. Mahalanobis, P. C. On the Generalized Distance in Statistics. *Proceedings of the National Institute of Sciences (Calcutta)* **2**, 49 (1936).
154. Malinen, M. I. & Fränti, P. in *Structural, Syntactic, and Statistical Pattern Recognition* (eds Fränti, P., Brown, G., Loog, M., Escolano, F. & Pelillo, M.) 32 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014).
155. Manen, S., Gygli, M., Dai, D. & Van Gool, L. *PathTrack: Fast Trajectory Annotation with Path Supervision* in 2017 IEEE International Conference on Computer Vision (ICCV) (IEEE, Venice, 2017), 290.
156. Marin, J., Lopez, A. M., Geronimo, D. & Vazquez, D. *Learning Appearance in Virtual Scenarios for Pedestrian Detection* in 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE Computer Society, Los Alamitos, CA, USA, 2010), 137.
157. Matthews, O., Ryu, K. & Srivastava, T. Creating a Large-Scale Synthetic Dataset for Human Activity Recognition. *arXiv preprint arXiv:2007.11118* (2020).
158. Mazzia, V., Angarano, S., Salvetti, F., Angelini, F. & Chiaberge, M. Action Transformer: A Self-Attention Model for Short-Time Pose-Based Human Action Recognition. *Pattern Recognition* **124**, 108487 (2022).
159. McGeoch, C., Farre, P. & Boothby, K. The D-Wave Advantage2 Prototype. *Technical Report* (2022).
160. McGeoch, C. & Farré, P. *The Advantage System: Performance Update* tech. rep. (D-Wave Systems Inc., 2021), 31.
161. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints* (2018).
162. Meinhardt, T., Kirillov, A., Leal-Taixe, L. & Feichtenhofer, C. *TrackFormer: Multi-Object Tracking with Transformers* in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE, New Orleans, LA, USA, 2022), 8834.
163. Meli, N. K., Mannel, F. & Lellmann, J. *An Iterative Quantum Approach for Transformation Estimation from Point Sets* in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022), 519.

164. Mercat, J., Gilles, T., El Zoghby, N., Sandou, G., Beauvois, D. & Gil, G. P. *Multi-Head Attention for Multi-Modal Joint Vehicle Motion Forecasting in 2020 IEEE International Conference on Robotics and Automation (ICRA) (2020)*, 9638.
165. Milan, A., Leal-Taixe, L., Reid, I., Roth, S. & Schindler, K. MOT16: A Benchmark for Multi-Object Tracking. *arXiv:1603.00831 [cs]* (2016).
166. Milan, A., Rezatofighi, S. H., Dick, A., Reid, I. & Schindler, K. *Online Multi-Target Tracking Using Recurrent Neural Networks in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)* (eds Singh, S. & Markovitch, S.) (Association for the Advancement of Artificial Intelligence (AAAI), United States of America, 2017), 4225.
167. Morris, B., Doshi, A. & Trivedi, M. *Lane Change Intent Prediction for Driver Assistance: On-road Design and Evaluation in 2011 IEEE Intelligent Vehicles Symposium (IV) (2011)*.
168. Mugel, S., Abad, M., Bermejo, M., Sánchez, J., Lizaso, E. & Orús, R. Hybrid Quantum Investment Optimization with Minimal Holding Period. *Scientific Reports* **11**, 19587 (2021).
169. Muhammad, K., Mustaqeem, Ullah, A., Imran, A. S., Sajjad, M., Kiran, M. S., Sannino, G. & de Albuquerque, V. H. C. Human Action Recognition Using Attention Based LSTM Network with Dilated CNN Features. *Future Generation Computer Systems* **125**, 820 (2021).
170. Mulligan, V. K., Melo, H., Merritt, H. I., Slocum, S., Weitzner, B. D., Watkins, A. M., Renfrew, P. D., Pelissier, C., Arora, P. S. & Bonneau, R. *Designing Peptides on a Quantum Computer* 2020.
171. Nelson, J., Vuffray, M., Likhov, A. Y., Albash, T. & Coffrin, C. High-Quality Thermal Gibbs Sampling with Quantum Annealing Hardware. *Physical Review Applied* **17**, 044046 (2022).
172. Neukart, F., Compostella, G., Seidel, C., von Dollen, D., Yarkoni, S. & Parney, B. Traffic Flow Optimization Using a Quantum Annealer. *Frontiers in ICT* **4**, 29 (2017).
173. Ngiam, J., Vasudevan, V., Caine, B., Zhang, Z., Chiang, H.-T. L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., Weiss, D. J., Sapp, B., Chen, Z. & Shlens, J. *Scene Transformer: A Unified Architecture for Predicting Future Trajectories of Multiple Agents in International Conference on Learning Representations (2022)*.
174. Nguyen, X. B., Thompson, B., Churchill, H., Luu, K. & Khan, S. U. *Quantum Vision Clustering* 2023.

175. Ohzeki, M., Miki, A., Miyama, M. J. & Terabe, M. Control of Automated Guided Vehicles Without Collision by Quantum Annealer and Digital Devices. *Frontiers in Computer Science* **1**, 9 (2019).
176. Paisitkriangkrai, S., Shen, C. & Van Den Hengel, A. *Learning to Rank in Person Re-Identification with Metric Ensembles in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE, Boston, MA, USA, 2015)*, 1846.
177. Pan, H., Zhu, X., Atici, S. & Cetin, A. E. *A Hybrid Quantum-Classical Approach Based on the Hadamard Transform for the Convolutional Layer in Proceedings of the 40th International Conference on Machine Learning 202 (JMLR.org, Honolulu, Hawaii, USA, 2023)*, 26891.
178. Pautrat, R., Lin, J.-T., Larsson, V., Oswald, M. R. & Pollefeys, M. *SOLD2: Self-supervised Occlusion-Aware Line Description and Detection in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)*, 11368.
179. Pellegrini, S., Ess, A., Schindler, K. & van Gool, L. *You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking in IEEE 12th International Conference on Computer Vision (IEEE, 2009)*, 261.
180. Peng, X., Usman, B., Saito, K., Kaushik, N., Hoffman, J. & Saenko, K. Syn2real: A New Benchmark For synthetic-to-Real Visual Domain Adaptation. *arXiv preprint arXiv:1806.09755* (2018).
181. Petsiuk, V., Das, A. & Saenko, K. *RISE: Randomized Input Sampling for Explanation of Black-Box Models in Proceedings of the British Machine Vision Conference (BMVC) (2018)*.
182. Plizzari, C., Cannici, M. & Matteucci, M. *Spatial Temporal Transformer Network for Skeleton-Based Action Recognition in Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part III (Springer, 2021)*, 694.
183. Pochart, T., Jacquot, P. & Mikael, J. *On the Challenges of Using D-Wave Computers to Sample Boltzmann Random Variables 2021*.
184. Poli, R., Kennedy, J. & Blackwell, T. M. Particle Swarm Optimization: An Overview. *Swarm Intelligence* **1** (2007).
185. Pollefeys, M., Van Gool, L. & Oosterlinck, A. *The Modulus Constraint: A New Constraint Self-Calibration in Proceedings of 13th International Conference on Pattern Recognition 1 (IEEE, 1996)*, 349.

186. Probst, T., Maninis, K.-K., Chhatkuli, A., Ourak, M., Vander Poorten, E. & Van Gool, L. Automatic Tool Landmark Detection for Stereo Vision in Robot-Assisted Retinal Surgery. *IEEE Robotics and Automation Letters* **3**, 612 (2017).
187. Qin, X., Cai, R., Yu, J., He, C. & Zhang, X. An Efficient Self-Attention Network for Skeleton-Based Action Recognition. *Scientific Reports* **12**, 4111 (2022).
188. Reid, D. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control* **24**, 843 (1979).
189. Rella, E. M., Zaech, J.-N., Liniger, A. & Van Gool, L. *Decoder Fusion RNN: Context and Interaction Aware Decoders for Trajectory Prediction in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021), 5937.
190. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 1137 (2017).
191. Ren, W., Wang, X., Tian, J., Tang, Y. & Chan, A. B. Tracking-by-Counting: Using Network Flows on Crowd Density Maps for Tracking Multiple Targets. *IEEE Transactions on Image Processing* **30**, 1439 (2021).
192. Ristani, E., Solera, F., Zou, R., Cucchiara, R. & Tomasi, C. *Performance Measures and a Data Set for Multi-target, Multi-camera Tracking in Computer Vision – ECCV 2016 Workshops* (eds Hua, G. & Jégou, H.) (Springer International Publishing, Cham, 2016), 17.
193. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. *High-Resolution Image Synthesis with Latent Diffusion Models in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New Orleans, LA, USA, 2022), 10674.
194. Rosenberg, A. & Hirschberg, J. *V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (Association for Computational Linguistics, Prague, Czech Republic, 2007), 410.
195. Roshan Zamir, A., Dehghan, A. & Shah, M. *GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs in Computer Vision – ECCV 2012* (eds Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y. & Schmid, C.) (Springer, Berlin, Heidelberg, 2012), 343.

196. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H. & Savarese, S. *SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints* in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Long Beach, CA, USA, 2019), 1349.
197. Salzmann, T., Ivanovic, B., Chakravarty, P. & Pavone, M. in *Computer Vision – ECCV 2020* (eds Vedaldi, A., Bischof, H., Brox, T. & Frahm, J.-M.) 683 (Springer International Publishing, Cham, 2020).
198. Sang, J., Wang, S. & Li, Q. A Novel Quantum Representation of Color Digital Images. *Quantum Information Processing* **16**, 42 (2016).
199. Saxena, A., Wong, L., Quigley, M. & Ng, A. Y. *A Vision-Based System for Grasping Novel Objects in Cluttered Environments* in *Robotics Research* (eds Kaneko, M. & Nakamura, Y.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011), 337.
200. Scaramuzza, D., Harati, A. & Siegwart, R. *Extrinsic Self Calibration of a Camera and a 3d Laser Range Finder from Natural Scenes* in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2007), 4164.
201. Schlechtriemen, J., Wedel, A., Hillenbrand, J., Breuel, G. & Kuhnert, K.-D. *A Lane Change Detection Approach Using Feature Ranking with Maximized Predictive Power* in *IEEE Intelligent Vehicles Symposium (IV)* (2014).
202. Schreier, M., Willert, V. & Adamy, J. *Bayesian, Maneuver-Based, Long-Term Trajectory Prediction and Criticality Assessment for Driver Assistance Systems* in *IEEE Conference on Intelligent Transportation Systems (ITSC)* (2014).
203. Schrödinger, E. Discussion of Probability Relations between Separated Systems. *Mathematical Proceedings of the Cambridge Philosophical Society* **31**, 555 (1935).
204. Shi, L., Wang, L., Zhou, S. & Hua, G. *Trajectory Unified Transformer for Pedestrian Trajectory Prediction* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 9675.
205. Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review* **41**, 303 (1999).

206. Simonyan, K. & Zisserman, A. *Two-Stream Convolutional Networks for Action Recognition in Videos* in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1* (MIT Press, Cambridge, MA, USA, 2014), 568.
207. Soomro, K., Zamir, A. R. & Shah, M. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv preprint arXiv:1212.0402* (2012).
208. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15**, 1929 (2014).
209. Su, J., Guo, X., Liu, C., Lu, S. & Li, L. An Improved Novel Quantum Image Representation and Its Experimental Test on IBM Quantum Experience. *Scientific Reports* **11**, 13879 (2021).
210. Su, T., Meng, Y. & Xu, Y. *Pedestrian Trajectory Prediction via Spatial Interaction Transformer Network* in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)* (IEEE, Nagoya, Japan, 2021), 154.
211. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z. & Anguelov, D. *Scalability in Perception for Autonomous Driving: Waymo Open Dataset* in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Seattle, WA, USA, 2020), 2443.
212. Sun, X. & Zheng, L. *Dissecting Person Re-Identification from the Viewpoint of Viewpoint* in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, Los Alamitos, CA, USA, 2019), 608.
213. Tan, M., Pang, R. & Le, Q. V. *EfficientDet: Scalable and Efficient Object Detection* in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Seattle, WA, USA, 2020), 10778.
214. Tang, S., Andriluka, M., Andres, B. & Schiele, B. *Multiple People Tracking by Lifted Multicut and Person Re-identification* in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Honolulu, HI, 2017), 3701.

215. Theagarajan, R. & Bhanu, B. An Automated System for Generating Tactical Performance Statistics for Individual Soccer Players from Videos. *IEEE Transactions on Circuits and Systems for Video Technology* **31**, 632 (2020).
216. Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. *Learning Spatiotemporal Features with 3D Convolutional Networks* in *2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, Santiago, Chile, 2015), 4489.
217. Trucco, E. & Verri, A. *Introductory Techniques for 3-D Computer Vision* (Prentice Hall Englewood Cliffs, 1998).
218. Unberath, M., Zaech, J.-N., Lee, S. C., Bier, B., Fotouhi, J., Armand, M. & Navab, N. *DeepDRR – A Catalyst for Machine Learning in Fluoroscopy-Guided Procedures in Medical Image Computing and Computer Assisted Intervention – MICCAI 2018* (eds Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C. & Fichtinger, G.) (Springer International Publishing, Cham, 2018), 98.
219. van Dam, W., Mosca, M. & Vazirani, U. *How Powerful Is Adiabatic Quantum Computation?* in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science* (2001), 279.
220. Vats, K., McNally, W., Walters, P., Clausi, D. A. & Zelek, J. S. *Ice Hockey Player Identification via Transformers and Weakly Supervised Learning in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 3451.
221. Vats, K., Walters, P., Fani, M., Clausi, D. A. & Zelek, J. S. *Player Tracking and Identification in Ice Hockey. Expert Systems with Applications* **213**, 119250 (2023).
222. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X. & Van Gool, L. *Temporal Segment Networks: Towards Good Practices for Deep Action Recognition in European Conference on Computer Vision* (Springer, 2016), 20.
223. Wang, L., Ran, Q., Ma, J., Yu, S. & Tan, L. *QRCI: A New Quantum Representation Model of Color Digital Images. Optics Communications* **438**, 147 (2019).
224. Wang, Y., Liang, X. & Liao, S. *Cloning Outfits from Real-World Images to 3D Characters for Generalizable Person Re-Identification* in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New Orleans, LA, USA, 2022), 4890.

225. Wang, Y., Liao, S. & Shao, L. *Surpassing Real-World Source Training Data: Random 3D Characters for Generalizable Person Re-Identification in Proceedings of the 28th ACM International Conference on Multimedia (Association for Computing Machinery, New York, NY, USA, 2020)*, 3422.
226. Wei, L., Zhang, S., Gao, W. & Tian, Q. *Person Transfer GAN to Bridge Domain Gap for Person Re-identification in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (IEEE, Salt Lake City, UT, USA, 2018)*, 79.
227. Weinland, D., Ronfard, R. & Boyer, E. Free Viewpoint Action Recognition Using Motion History Volumes. *Computer vision and image understanding* **104**, 249 (2006).
228. Wen, F., Li, M. & Wang, R. *Social Transformer: A Pedestrian Trajectory Prediction Method Based on Social Feature Processing Using Transformer in 2022 International Joint Conference on Neural Networks (IJCNN) (2022)*, 1.
229. Weng, X., Wang, J., Held, D. & Kitani, K. *3D Multi-Object Tracking: A Baseline and New Evaluation Metrics in Proceedings of (IROS) IEEE/RSJ International Conference on Intelligent Robots and Systems (2020)*.
230. Weng, X., Wang, Y., Man, Y. & Kitani, K. M. *GNN₃DMOT: Graph Neural Network for 3D Multi-Object Tracking With 2D-3D Multi-Feature Learning in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE, Seattle, WA, USA, 2020)*, 6498.
231. Wiebe, N., Kapoor, A. & Svore, K. M. Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning. *Quantum Info. Comput.* **15**, 316 (2015).
232. Wilde, M. M. *Quantum Information Theory* 2nd ed. (Cambridge University Press, Cambridge, 2017).
233. Wojke, N. & Bewley, A. *Deep Cosine Metric Learning for Person Re-Identification in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (IEEE, 2018)*, 748.
234. Wojke, N., Bewley, A. & Paulus, D. *Simple Online and Realtime Tracking with a Deep Association Metric in 2017 IEEE International Conference on Image Processing (ICIP) (IEEE, 2017)*, 3645.

235. Wright, K., Beck, K. M., Debnath, S., Amini, J. M., Nam, Y., Grzesiak, N., Chen, J.-S., Pisenti, N. C., Chmielewski, M., Collins, C., Hudek, K. M., Mizrahi, J., Wong-Campos, J. D., Allen, S., Apisdorf, J., Solomon, P., Williams, M., Ducore, A. M., Blinov, A., Kreikemeier, S. M., Chaplin, V., Keesan, M., Monroe, C. & Kim, J. Benchmarking an 11-Qubit Quantum Computer. *Nature Communications* **10**, 5464 (2019).
236. Wu, Y., Bao, W.-S., Cao, S., Chen, F., Chen, M.-C., Chen, X., Chung, T.-H., Deng, H., Du, Y., Fan, D., Gong, M., Guo, C., Guo, C., Guo, S., Han, L., Hong, L., Huang, H.-L., Huo, Y.-H., Li, L., Li, N., Li, S., Li, Y., Liang, F., Lin, C., Lin, J., Qian, H., Qiao, D., Rong, H., Su, H., Sun, L., Wang, L., Wang, S., Wu, D., Xu, Y., Yan, K., Yang, W., Yang, Y., Ye, Y., Yin, J., Ying, C., Yu, J., Zha, C., Zhang, C., Zhang, H., Zhang, K., Zhang, Y., Zhao, H., Zhao, Y., Zhou, L., Zhu, Q., Lu, C.-Y., Peng, C.-Z., Zhu, X. & Pan, J.-W. Strong Quantum Computational Advantage Using a Superconducting Quantum Processor. *Physical Review Letters* **127**, 180501 (2021).
237. Wu, Y., Bao, W.-S., Cao, S., Chen, F., Chen, M.-C., Chen, X., Chung, T.-H., Deng, H., Du, Y., Fan, D., Gong, M., Guo, C., Guo, C., Guo, S., Han, L., Hong, L., Huang, H.-L., Huo, Y.-H., Li, L., Li, N., Li, S., Li, Y., Liang, F., Lin, C., Lin, J., Qian, H., Qiao, D., Rong, H., Su, H., Sun, L., Wang, L., Wang, S., Wu, D., Xu, Y., Yan, K., Yang, W., Yang, Y., Ye, Y., Yin, J., Ying, C., Yu, J., Zha, C., Zhang, C., Zhang, H., Zhang, K., Zhang, Y., Zhao, H., Zhao, Y., Zhou, L., Zhu, Q., Lu, C.-Y., Peng, C.-Z., Zhu, X. & Pan, J.-W. Strong Quantum Computational Advantage Using a Superconducting Quantum Processor. *Physical Review Letters* **127**, 180501 (2021).
238. Xu, R. & Wunsch, D. Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks* **16**, 645 (2005).
239. Xu, Y., sep, A., Ban, Y., Horaud, R., Leal-Taixe, L. & Alameda-Pineda, X. *How to Train Your Deep Multi-Object Tracker in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Seattle, WA, USA, 2020), 6786.
240. Yamamoto, T., Kataoka, H., Hayashi, M., Aoki, Y., Oshima, K. & Tanabiki, M. *Multiple Players Tracking and Identification Using Group Detection and Player Number Recognition in Sports Video in IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society* (IEEE, 2013), 2442.

241. Yang, J., Parikh, D. & Batra, D. *Joint Unsupervised Learning of Deep Representations and Image Clusters* in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Las Vegas, NV, USA, 2016), 5147.
242. Yang, J., Ge, H., Yang, J., Tong, Y. & Su, S. *Online Multi-Object Tracking Using Multi-Function Integration and Tracking Simulation Training*. *Applied Intelligence* (2021).
243. Yang, Y. & Soatto, S. *FDA: Fourier Domain Adaptation for Semantic Segmentation* in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Seattle, WA, USA, 2020), 4084.
244. Yin, T., Zhou, X. & Krähenbühl, P. *Center-Based 3D Object Detection and Tracking* in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 11779.
245. Yin, Z., Liu, R., Xiong, Z. & Yuan, Z. *Multimodal Transformer Networks for Pedestrian Trajectory Prediction* in *Twenty-Ninth International Joint Conference on Artificial Intelligence 2* (2021), 1259.
246. Yurtsever, A., Birdal, T. & Golyanik, V. *Q-FW: A Hybrid Classical-Quantum Frank-Wolfe for Quadratic Binary Optimization* in *Computer Vision – ECCV 2022* (eds Avidan, S., Brostow, G., Cissé, M., Farinella, G. M. & Hassner, T.) (Springer Nature Switzerland, Cham, 2022), 352.
247. Zaech, J.-N., Dai, D., Hahner, M. & Gool, L. V. *Texture Underfitting for Domain Adaptation* in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019), 547.
248. Zaech, J.-N., Dai, D., Liniger, A. & Van Gool, L. *Action Sequence Predictions of Vehicles in Urban Environments Using Map and Social Context* in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Las Vegas, NV, USA, 2020), 8982.
249. Zaech, J.-N., Gao, C., Bier, B., Taylor, R., Maier, A., Navab, N. & Unberath, M. *Learning to Avoid Poor Images: Towards Task-aware C-arm Cone-beam CT Trajectories in Medical Image Computing and Computer Assisted Intervention – MICCAI 2019* (eds Shen, D., Liu, T., Peters, T. M., Staib, L. H., Essert, C., Zhou, S., Yap, P.-T. & Khan, A.) (Springer International Publishing, Cham, 2019), 11.
250. Zaech, J.-N., Liniger, A., Danelljan, M., Dai, D. & Van Gool, L. *Adiabatic Quantum Computing for Multi Object Tracking* in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New Orleans, LA, USA, 2022), 8801.

251. Zhai, X., Hu, Z., Yang, D., Zhou, L. & Liu, J. *Social Aware Multi-modal Pedestrian Crossing Behavior Prediction in Computer Vision – ACCV 2022* (eds Wang, L., Gall, J., Chin, T.-J., Sato, I. & Chellappa, R.) (Springer Nature Switzerland, Cham, 2023), 275.
252. Zhang, C. & Berger, C. *Learning the Pedestrian-Vehicle Interaction for Pedestrian Trajectory Prediction in* (2022), 230.
253. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y. & Wen, F. *Prototypical Pseudo Label Denoising and Target Structure Learning for Domain Adaptive Semantic Segmentation in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Nashville, TN, USA, 2021), 12409.
254. Zhang, W., Zhou, H., Sun, S., Wang, Z., Shi, J. & Loy, C. C. *Robust Multi-Modality Multi-Object Tracking in 2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, Seoul, Korea (South), 2019), 2365.
255. Zhang, Y., Lu, K., Gao, Y. & Wang, M. NEQR: A Novel Enhanced Quantum Representation of Digital Images. *Quantum Information Processing* **12**, 2833 (2013).
256. Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Transactions on pattern analysis and machine intelligence* **22**, 1330 (2000).
257. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J. & Tian, Q. *Scalable Person Re-identification: A Benchmark in 2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, Santiago, Chile, 2015), 1116.
258. Zheng, Z., Yang, X., Yu, Z., Zheng, L., Yang, Y. & Kautz, J. *Joint Discriminative and Generative Learning for Person Re-Identification in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Long Beach, CA, USA, 2019), 2133.
259. Zhong, H.-S., Deng, Y.-H., Qin, J., Wang, H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Wu, D., Gong, S.-Q., Su, H., Hu, Y., Hu, P., Yang, X.-Y., Zhang, W.-J., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., Wang, Z., Li, L., Liu, N.-L., Renema, J. J., Lu, C.-Y. & Pan, J.-W. Phase-Programmable Gaussian Boson Sampling Using Stimulated Squeezed Light. *Physical Review Letters* **127**, 180502 (2021).
260. Zhou, X., Koltun, V. & Krähenbühl, P. *Tracking Objects as Points in Computer Vision – ECCV 2020* (eds Vedaldi, A., Bischof, H., Brox, T. & Frahm, J.-M.) (Springer International Publishing, Cham, 2020), 474.

261. Zhu, B., Jiang, Z., Zhou, X., Li, Z. & Yu, G. Class-Balanced Grouping and Sampling for Point Cloud 3D Object Detection. *arXiv:1908.09492 [cs]* (2019).

PUBLICATIONS

Articles in peer-reviewed journals:

1. Unberath*, M., **J.-N. Zaech***, Gao*, C., Bier, B., Goldmann, F., Lee, S. C., Fotouhi, J., Taylor, R., Armand, M. & Navab, N. Enabling machine learning in X-ray-based procedures via realistic simulation of image formation. *International Journal of Computer Assisted Radiology and Surgery* (2019).
* Joint first authors.
2. Thies*, M., **J.-N. Zaech***, Gao, C., Taylor, R., Navab, N., Maier, A. & Unberath, M. A Learning-based Method for Online Adjustment of C-arm Cone-Beam CT Source Trajectories for Artifact Avoidance. *International Journal of Computer Assisted Radiology and Surgery* (2020).
* Joint first authors listed alphabetically.
3. Herl, G., Hiller, J., Thies, M., **J.-N. Zaech**, Unberath, M. & Maier, A. Task-Specific Trajectory Optimisation for Twin-Robotic X-Ray Tomography. *IEEE Transactions on Computational Imaging* 7 (2021).
4. Bier, B., Goldmann, F., **J.-N. Zaech**, Fotouhi, J., Hegeman, R., Grupp, R., Armand, M., Osgood, G., Navab, N., Maier, A. & Unberath, M. Learning to detect anatomical landmarks of the pelvis in X-rays from arbitrary views. *International Journal of Computer Assisted Radiology and Surgery* (2019).
5. **J.-N. Zaech**, Liniger, A., Dai, D., Danelljan, M. & Van Gool, L. Learnable Online Graph Representations for 3D Multi-Object Tracking. *IEEE Robotics and Automation Letters* 7, 5103 (2022).

Conference contributions:

1. **J.-N. Zaech**, Gao, C., Bier, B., Taylor, R., Maier, A., Navab, N. & Unberath, M. *Learning to Avoid Poor Images: Towards Task-aware C-arm Cone-beam CT Trajectories in Medical Image Computing and Computer Assisted Intervention* (2019).
2. **J.-N. Zaech**, Dai, D., Hahner, M. & Van Gool, L. *Texture Underfitting for Domain Adaptation in IEEE Intelligent Transportation Systems Conference* (2019).

3. **J.-N. Zaech**, Dai, D., Liniger, A. & Van Gool, L. *Action Sequence Predictions of Vehicles in Urban Environments using Map and Social Context in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020).
4. **J.-N. Zaech**, Liniger, A., Danelljan, M., Dai, D. & Van Gool, L. *Adiabatic Quantum Computing for Multi Object Tracking in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)* (IEEE, New Orleans, LA, USA, 2022).
5. **J.-N. Zaech**, Danelljan, M., Birdal, T. & Van Gool, L. *Probabilistic Sampling of Balanced K-Means Using Adiabatic Quantum Computing in IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Seattle, WA, USA, 2024).
6. Unberath*, M., **J.-N. Zaech***, Lee, S. C., Bier, B., Fotouhi, J., Armand, M. & Navab, N. *DeepDRR – A Catalyst for Machine Learning in Fluoroscopy-Guided Procedures in Medical Image Computing and Computer Assisted Intervention* (2018), 98.
* Joint first authors listed alphabetically.
7. Mello Rella, E., **J.-N. Zaech**, Liniger, A. & Van Gool, L. *Decoder Fusion RNN: Context and Interaction Aware Decoders for Trajectory Prediction in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021).
8. Agarwal, P., Pani Paudel, D., **J.-N. Zaech** & Van Gool, L. *Unsupervised Robust Domain Adaptation without Source Data in 2022 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2021).
9. Albanese*, G., Mitra*, A., **J.-N. Zaech***, Zhao*, Y., Chhatkuli, A. & Van Gool, L. *Optimizing Long-Term Player Tracking and Identification in NAO Robot Soccer by fusing Game-state and External Video in IEEE Winter Conference on Applications of Computer Vision (WACV)* (IEEE, 2024).
* Joint first authors listed alphabetically.
10. Bier*, B., Unberath*, M., **J.-N. Zaech**, Fotouhi, J., Armand, M., Osgood, G., Navab, N. & Maier, A. *X-ray-transform Invariant Anatomical Landmark Detection for Pelvic Trauma Surgery in Medical Image Computing and Computer Assisted Intervention* (2018), 55.
* Joint first authors listed alphabetically.
11. Hahner, M., Dai, D., Sakaridis, C., **J.-N. Zaech** & Van Gool, L. *Semantic Understanding of Foggy Scenes with Purely Synthetic Data in IEEE Intelligent Transportation Systems Conference* (2019).

12. Mitra*, A., Molnar*, L., **J.-N. Zaech***, Wu, Y., Oliveira, C., Heo, S., Fisher, Y. & Van Gool, L. *Multi-Domain Referee Dataset: Enabling Recognition of Referee Signals on Robotic Platforms in IROS Human Multi-Robot Interaction* (IEEE, 2023).

* Joint first authors listed alphabetically.

13. Schram, V., Bereyhi, A., **J.-N. Zaech**, Müller, R. R. & Gerstacker, W. H. *Approximate Message Passing for Indoor THz Channel Estimation in International Balkan Conference on Communications and Networking* (2019).