Diss. ETH No. 28807

Accelerating Molecular Discovery with Generative Language Models

A journey through the chemical space

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. sc. ETH Zürich)

presented by

Jannis Born
M.Sc., UZH ETH

born on 20.03.1994
citizen of Germany

accepted on the recommendation of:

Prof. Dr. Karsten Borgwardt, ETH Zurich, examiner
Dr. Matteo Manica, IBM Research Europe, co-examiner
Prof. Dr. Alán Aspuru-Guzik, University of Toronto, co-examiner

2022

Abstract

The discovery of new molecules and materials with desired properties is pivotal to our success in combatting global challenges such as the climate crisis or emerging diseases. However, navigating the discrete and practically infinite chemical search space while having to respect a cascade of multiproperty objectives is extremely challenging. In the past few decades, the chemical industry has faced not only a decline in productivity, but also ever-rising costs for the research and development of novel materials and molecules.

Recently, molecular generative models coupled with virtual screening methods have shown promising results in efficient and systematic chemical space exploration. The hopes are high that such methods can accelerate the molecular discovery process, in particular when coupled with chemical synthesis planning tools and robotic hardware in automated laboratories. However, most generative models are optimized toward simplistic, chemocentric objectives, disregard system-level information about the target environment of the molecule and can thus not be applied to generate molecules *conditionally* for a wide range of objectives.

This thesis is about developing conditional molecular generative models that can be queried with a semantic context and flexibly generate molecules for desired conditions without the need of specific optimization. Moreover, this thesis aims to improve the "entanglement" of *de novo* design and property prediction by developing molecular generative models that possess inductive biases about continuous properties and also excel at predicting such properties. This is achieved by exploiting analogies between natural language and organic chemistry.

As a prerequisite for generative modeling, the first part of this thesis is devoted to building predictive models for molecular properties. The first chapter presents a simple, yet robust and interpretable chemical language model that heavily relies on data augmentation and is shown to exhibit strong performance across a wide range of properties such as toxicity. The next chapter develops proteochemometric language models for protein-ligand binding affinity prediction and demonstrates that by discarding more than 95% of the residues from the protein sequence, the performance of binding affinity prediction for human protein kinases significantly *improves*.

The second part of this thesis focuses on the main goal of developing generative language models for conditional molecular design. Leveraging the property predictors in a reinforcement-learning optimization scheme yields a generative model that can be conditioned on a biomolecular context vector (e.g., a gene expression signature of a malignant tumour or a target protein) and generate molecules with high affinity toward this context. The experiments show that this method generalizes well and can propose molecules with high selectivity for unseen protein targets even in the absence of experimental data for

such targets. In a case study on accelerated molecular discovery, the proposed generative model is integrated into a completely autonomous workflow that spans retrosynthesis models, synthesis protocol generation and the successful wet-lab synthesis on a robotic hardware.

The last chapter then proposes a multitask language model that abstracts regression as a conditional sequence modeling problem and thus unifies the previous work on molecular property prediction and conditional generation within the same model. This model not only excels on regression tasks despite relying on a classification loss, it can also be conditioned simultaneously on arbitrary molecular substructures and continuous target properties. As demonstrated, this model outperforms specialized approaches in conditional molecular design and can decorate seed molecules, proteins or chemical reactions based on a desired property primer without the need of any optimization. This finds particular application in property-driven local exploration of the chemical space and paves the road toward foundation models in material design.

Altogether, this thesis may contribute toward accelerated molecular discovery by providing methods to improve the quality of the average hypothesis that is considered for downstream chemical synthesis and wet-lab experimentation.

Zusammenfassung

Die Entdeckung neuer Moleküle und Materialien mit gewünschten Eigenschaften wird entscheidend sein für unseren Erfolg bei der Bekämpfung globaler Herausforderungen wie der Klimakrise oder neu auftretender Krankheiten. Den diskreten und praktisch unendlich grossen chemischen Suchraum unter einer Kaskade von Mehrzieloptimierung zu navigieren ist jedoch eine grosse Herausforderung. In den letzten Jahrzehnten sah sich die chemische Industrie nicht nur mit einem Produktivitätsrückgang, sondern auch mit ständig steigenden Kosten für die Erforschung und Entwicklung neuer Materialien und Moleküle konfrontiert.

In den letzten Jahren haben molekulare generative Modelle in Verbindung mit virtuellen Screening-Methoden vielversprechende Ergebnisse zur effizienteren und systematischeren Erkundung des chemischen Raums gezeigt. Die Hoffnungen sind gross, dass solche Methoden den molekularen Entdeckungsprozess beschleunigen können, insbesondere wenn sie mit Tools zur Planung der chemischen Synthese und Roboterhardware in automatisierten Labors gekoppelt werden. Die meisten generativen Modelle optimieren jedeoch vereinfachte, chemozentrische Ziele, vernachlässigen system-biologische Informationen über die die Zielumgebung des Moleküls und können daher nicht zur Generierung von Molekülen für ein breites Spektrum von Zielen verwendet werden.

In dieser Dissertation entwickeln wir konditionale molekulare generative Modelle, die basierend auf semantischen Kontext flexibel Moleküle generieren, ohne dass eine spezifische Optimierung erforderlich ist. Wir bedienen uns dafür Analogien zwischen natürlicher Sprache und organischer Chemie und verwenden Methoden der Sprachverarbeitung, insbesondere um generative Modelle besser mit Modellen zur Eigenschaftsvorhersage zu verknüpfen.

Um den Grundstein für die später entwickelten generativen Modelle zu legen, widmet sich der erste Teil der Dissertation zunächst der Erstellung von Vorhersagemodellen für molekulare Eigenschaften. Wir präsentieren ein einfaches, aber robustes und interpretierbares chemisches Sprachmodell, das in hohem Maße auf Augmentation der Daten beruht und, wie wir nachweisen, eine hohe Treffsicherheit in der Vorhersage eines breiten Spektrums von molekularen Eigenschaften wie Toxizität aufweist. Anschließend entwickeln wir mehrere proteochemometrisches Sprachmodelle für die Vorhersage von Protein-Liganden-Bindungsaffinität. Wir zeigen, dass die Präzision der Bindungsaffinitätsvorhersage für menschliche Proteinkinasen erheblich verbessert werden kann, wenn nur weniger als 5% der Proteinresiduen genutzt werden.

Im zweiten Teil dieser Dissertation konzentrieren wir uns auf das Hauptziel: konditionale generative Sprachmodelle zum Moleküldesign zu entwickeln. Zunächst nutzen wir die vorher entwickelten Eigenschaftsprädiktoren in einem Optimierungsschema mit Reinforcement Learning, um einen Molekülgenerator zu entwickeln, der auf einen

biomolekularen Kontextvektor (z. B. eine Genexpressionssignatur eines Tumors oder ein Zielprotein) konditioniert werden kann und Moleküle mit hoher Affinität zu diesem Kontext erzeugt. Wir stellen fest, dass diese Methode gut generalisiert ist und Moleküle mit hoher Selektivität für unbekannte Proteinziele vorschlagen kann, selbst wenn keine experimentellen Daten von diesen Proteinen zum Trainieren des Modells vorliegen. In einer Fallstudie zur beschleunigten molekularen Entdeckung integrieren wir unser generatives Modell in einen vollständig autonomen Arbeitsablauf. Dieser umfasst Retrosynthesemodelle, die automatische Generierung von Syntheseprotokollen und einen chemischen Laborrobotor, der unser Zielmolekül ohne menschlische Unterstützung synthetisiert.

Im letzten Kapitel schlagen wir dann ein Multitasking-Sprachmodell vor. Indem wir ein Regressionsproblem als bedingtes Sequenzmodellierungsproblem formulieren, vereinigen wir unsere bisherigen Beiträge zur Vorhersage molekularer Eigenschaften und zur konditionalen Generierung innerhalb *eines* Modells. Dieses Modell eignet sich nicht nur hervorragend für Regressionsaufgaben (obwohl es auf einer klassifikations-basierten Optimierungsfunktion beruht) sondern es kann auch gleichzeitig auf beliebige molekulare Unterstrukturen und kontinuierliche Zieleigenschaften konditioniert werden. Wie wir zeigen, übertrifft dieses Modell spezialisierte Ansätze im Bereich des konditionalen Moleküldesigns und kann Ausgangsmoleküle, Proteine oder chemische Reaktionen auf der Grundlage einer präzise gewählten, gewünschten Eigenschaft anpassen, ohne dass eine explizite Optimierung erforderlich ist. Diese Methode findet insbesondere bei der eigenschaftsgesteuerten lokalen Erkundung des chemischen Raums Anwendung und ebnet den Weg für foundation models im Materialdesign.

Insgesamt hoffen wir, dass diese Arbeit einen geringen Beitrag zur beschleunigten molekularen Entdeckung leisten kann, indem sie Methoden vorstellt, die es ermöglichen die durchschnittliche Qualität eines neuen Moleküls, das für Laborsynthese und Laborexperimente in Betracht gezogen wird, zu verbessern.

Acknowledgments

This thesis would not exist without all the people that backed me personally, educated me scientifically and facilitated my collaborative research in the last three years. I have been amazingly privileged to have had the opportunity to conduct this PhD jointly between IBM Research and ETH Zurich. The trust that was put in me has always been one of my biggest motivators.

First and foremost I would like to express my deepest gratitude to Matteo, my supervisor. It is hard to put in words how vital you have been for my work. Your mental sharpness, your stamina and your drive are truly admirable. But learning so much from you and having so many doors opened have been the real gifts. You have been a role model on so many aspects that, simply put, I could not have imagined a better supervisor.

I would like to extend my gratitude to my advisor Prof. Dr. Karsten Borgwardt for providing me the opportunity to do my PhD, for guidance and valuable scientific advice but also for the freedom to explore my own ideas. I am especially thankful for your unconditional and intensive support in the challenging times.

I am truly honoured to have worked with so many great researchers and great people at IBM Research. Teo, I am deeply thankful for your trust and the opportunity to work in your team and I am extremely excited to write the next chapter together. I admire you for being a visionary and a true leader, for all you pioneering work and just for your ability to make things happen. Joris, thank you for all the work we did together since the first day, for educating me on DevOps and for always having a joke on your lips. I thank Maria Gabrani, my former manager, for her trust to let me join her team, for her guidance and for teaching me what it means to be an IBMer. To my officemates: Giorgio, working with you is a blast, but the fun we had in the office and the philosophical discussions about virtually anything were even better. Anna, supporting you to raise your project from a baby to a TITAN has been a real pleasure, it is a pity we dont see each other more often. Aurelien, you always managed to surprise me – after all, I had a lot of fun. Thank you Marianna and Antonio for entertaining the lunches back in the day, Marianna for giving me a tremendous amount of advice and Antonio for sharing the excitement about the latest Strava activities. To the fellow PhD students: Thanks Dimitrios for being my ZHC2 and CCC expert, Oliver for all the shoptalking about outdoor stuff, Alessandra, for always spreading your positive energy through the group; Yves, for saying "Good morning!" at 2pm which always helped me to stop feeling guilty to sleep in; and – of course –, Alek: it is difficult to find someone more humble than you. Thanks to all the incredibly talented master students I had the luck to work with closely, Greta, Nikita and, also: Nil – it is great to have you at Arctoris now. Nikita special thanks to you for becoming a friend, for your endurance and for always spreading your energy and cheerfulness. Alain, Amol, I am learning so much from you and I find it truly inspiring how you mastered both chemistry and computer science. Federico, I simply wish I had your

energy level. Philippe, even though we never really worked together, you inspired a lot of my work. Thanks to Maria Rodriguez for her guidance and her support in following my own ideas. Ali for paving the road for me to join IBM Research, for being a great advisor and an entertaining officemate. Carlo, Artem, Oliver, Alessandro and Antonio Cardinale not only for all the hard work in the lab on our projects but also for reading some meaning into the junk from my generative models. I am very thankful also to all the others in the team, Miruna, Anuv, Veronika, Astrid – for a short but very effective collaboration –, Peter, Nicolai, Adrian and especially Mark, for giving gediatric care to my old bike. Thanks also to Adriano, Alice, Nicolas, Andrea, Francesco, Kevin, Panos, Pushpak and An-Phi for insightful and entertaining discussions and to all the other master students I interacted with, Modestas, Antonio Berros, Caner, Patricia, Zoe, George, Nikos, Pedro and Samuel. All of you made my life and work in the lab extremely pleasing.

I am also thankful to all the IBM Researchers from abroad: To the polymer team in Almaden: Dmitry, Nate, Jim and Tim for educating me in our biweekly meetings that always left me with googling some stuff to catch up. To Yoel in Haifa, for many great brainstorming sessions and always having an open ear. To Girmaw, Celia and William in Nairobi for your drive to jump into a new domain together. To Wendy and Tienh in Yorktown about the opportunity to collaborate. Michal Rosen-Zvi and Maria Gabrani for onboarding me for my interlude about medical image analysis and COVID-19, the leadership throughout the project and granting me the responsibility to play a key role. Deepta Rajan, David Beymer and the countless collaborators of that project who made it a really an interdisciplinary community effort in the midst of the pandemic.

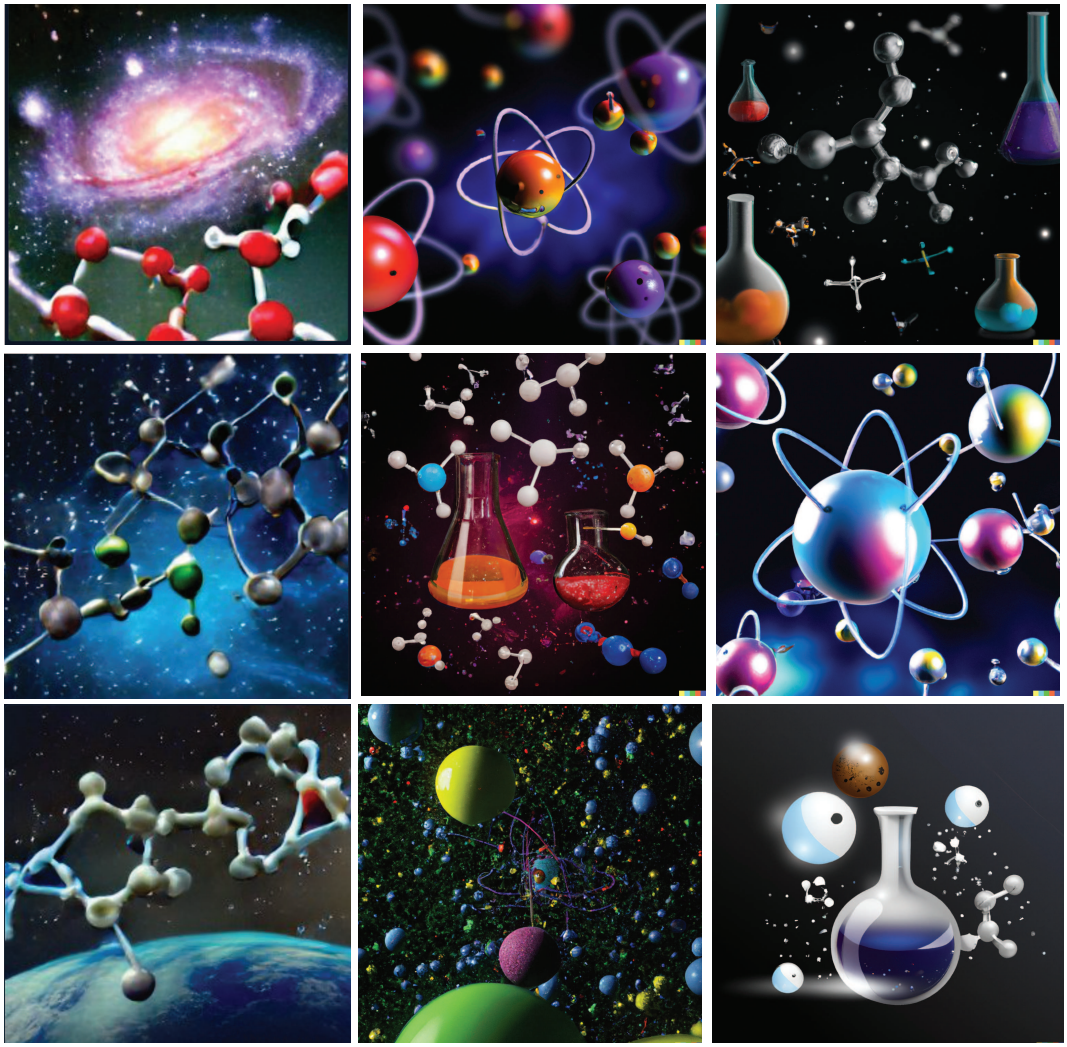
I am grateful to Prof. Dr. Alán Aspuru-Guzik, for his time to serve in my PhD committee, for always creating pleasant atmospheres and, of course, for inspiring research.

Next, I would like to express my gratitude to all the former and current MLCB members I had the pleasure to interact with (Tae-Hoon, Leslie, Dexiong, Michael, Tim and Max). Most importantly: Bastian, I always valued your advice and over the years, you became a really important mentor for me. Working with your LaTeX-`mimosis` template simply has been a blast. Thanks to Andrea Volkamer and Talia Kimber for a nice collaboration and for keep pushing our project forward.

I am also extremely thankful to all the anonymous developers of open-source software, for example, the inventors of AutoGrad without whom I would still be miserably failing trying to derive gradients myself. To name a few explicitly, to Greg Landrum for RDKit, Daniel Probst for TMap/Faerun, Mario Krenn for SELFIES and of course, everybody who developed the stellar IBM RXN for Chemistry.

It goes without saying that I am forever grateful for all my supportive friends and family members. Lukas, you always help recalibrating me on what truly counts in life and your ability to read the soul of humans is truly unique. To my father Michael and my sister Ronja, I feel your ubiquitous support in whatever I do and I owe you everything.

Last, Nina, no words can describe what I feel for you. You are my mental twin and I truly hope that you remain the most important person I will ever meet in my life.



"A journey through the chemical space."
— DALL-E2 2022.

Contents

1 Introduction

1.1 The need to accelerate molecular discovery

Although technological advances and high-throughput screenings are revolutionizing our understanding of basic biological and chemical processes, the chemical industry is facing ever-rising costs in research and development to bring new materials and pharmaceuticals to the market. Traditional approaches are largely driven by human hypotheses which are inevitably biased, ad-hoc and non-exhaustive. One particularly affected field is the pharmaceutical industry. Reversed to *Moore*'s law, the term *Eroom*'s law was coined for the observation that the number of FDA-approved drugs per billion invested US$ is halving every nine years since around 1950 [1]. To date, the estimated costs per new drug accumulate to up to 3B US$ [2]. The reasons for these ever-rising costs are numerous but certainly include

- the **attrition rate**. While throughout the history only ∼1,500 compounds received FDA-approval [3], the total number of ever synthesized and researched molecules is at least 60M [4]. Hence, it is not surprising that the success rate from *in vitro* screening to release on market has been estimated to be $< 0.01\%$ [1].
- the **serendipity**. More than 5% of all marketed pharmaceuticals profited from coincidental findings [5].
- the **wariness**. It usually takes 10-15 years from hit identification to market approval, with almost 10 years spent in the cascade of clinical trials and safety checks [6].
- the **search space**. The pharmaceutically promising part of the chemical space is practically infinite – it has been estimated to contain $\sim 10^{30}$-10^{60} molecules [7]. Only a tiny fraction of that space ($< 10^8$) has been explored thus far.

Molecular discovery can be seen as a multiproperty optimization in a discrete search space that is practically infinitely large. Due to this complexity, the classical and ubiquitous scientific method of empirical knowledge acquisition, that has been proven indispensable for modern science is reaching its limits. This scientific method is visualized in Figure 1.1**A** and includes a cycle of raising a *question* (i.e., identifying a problem), *studying* the problem, developing a *hypothesis* to answer the question, *testing* the hypothesis, *analyzing* the result and, finally, adapting the *question* based on the collected evidence.

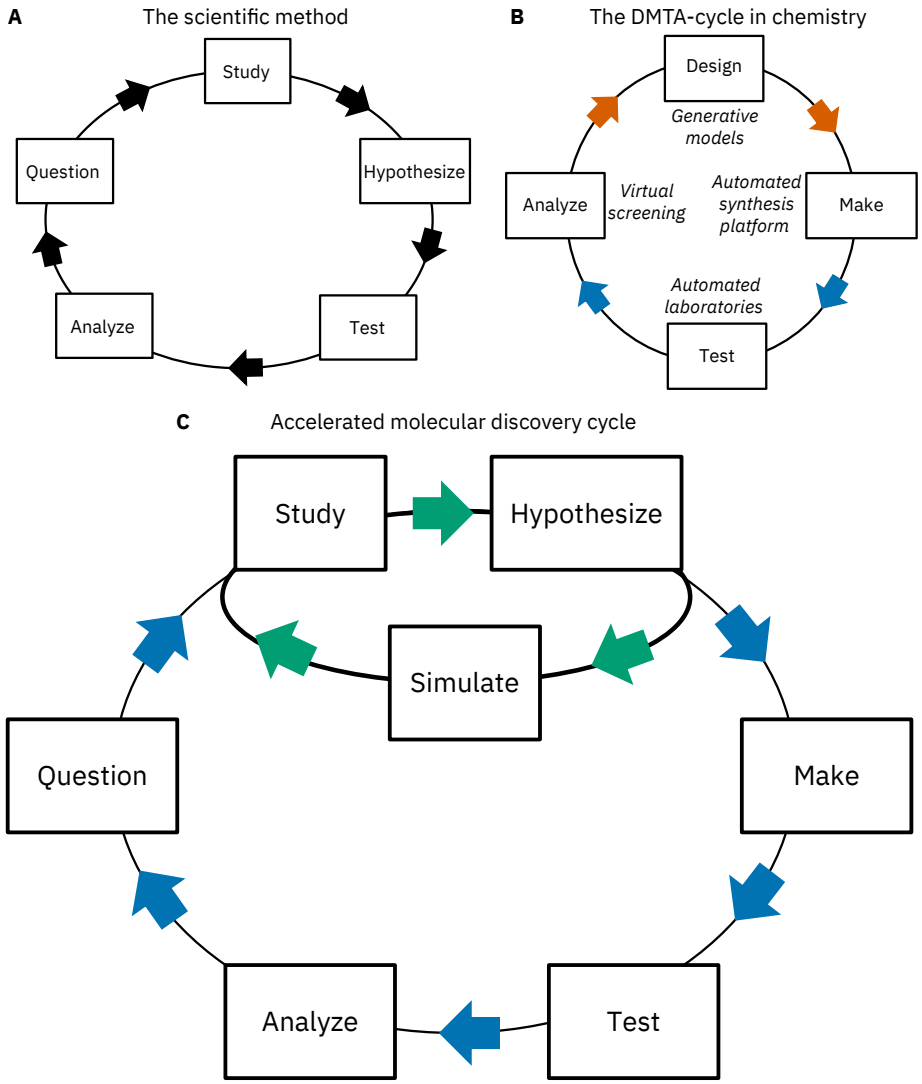


Figure 1.1: Accelerating molecular discovery. A) The classical scientific method of empirical knowledge acquisition is often seen as a cycle of raising a question, studying the existing evidence, formulating a hypothesis, testing it empirically and analyzing the results. **B)** In chemistry, this procedure is aptly summarized in the design-make-test-analyse (DMTA) cycle which, critically, also involves *making* the molecule in the lab. The bottlenecks of the cycle are highlighted with red arrows. Inside the circle, we name the needed technologies to automatize the DMTA cycle. **C)** The accelerated molecular discovery cycle. Generative models promise to support the hypothesize/design step of the DMTA cycle. Critical aspects are 1) controlling generative models to generate better, more targeted hypotheses and 2) evaluating the generated hypotheses via simulations *before* the costly synthesis and test phases are entered. This validation loop can be seen as an additional small cycle, embedded into the canonical DMTA cycle. This cycle is marked with green arrows and will be the focus of this thesis.

In the field of chemistry, the scientific method is commonly summarized in the Design-Make-Test-Analyze (DMTA) cycle which, critically, also involves the *make* of the molecule via chemical synthesis (cf. Figure 1.1**B**). This step is costly and time-consuming – consequently "closing the loop" to refine the hypothesis becomes substantially harder which introduces a bottleneck to the discovery cycle. Taken together, the poor quality of the *hypotheses* (usually proposed by humans in an ad-hoc manner) in combination with the time/cost bottleneck to *test* the hypotheses are responsible for the current molecular discovery crisis.

To combat global challenges such as climate change or emerging diseases, an accelerated form of molecular discover could become pivotal. *Tabor* et al. [8] proclaimed that this necessitates the integration of four technologies, namely (1) high-throughput virtual screening, (2) ML algorithms, especially molecular generative models, (3) tools for automated synthesis planning and (4) automated laboratories. From these four aspects, we believe that the key challenge in order to reduce the bottleneck in molecular discovery will be to improve the quality of an average hypothesis that is evaluated in the lab. Recently, deep generative models have emerged as a promising tool to expedite the hypothesis/design step in molecular discovery [9, 10]. The hopes are high that these models allow to explore the chemical space more systematically and on a larger scale [11]. However, even the best molecular generative model necessitates a counterpart – an efficient method for large-scale virtual screening that allows to *test* the generated hypotheses. This interplay can best be summarized in a new paradigm shown in Figure 1.1**C**, the "accelerated molecular discovery cycle". In this cycle, an additional small validation loop is embedded into the canonical DMTA cycle. This validation loop entails the advantages that a large number of generated hypotheses can be evaluated rapidly and at almost no cost. In addition, this loop can improve the generative model employed for the design phase and thus ensure that only the best hypotheses proceed to the synthesis and physical experimentation stage. Clearly, the success of this inner cycle not only depends upon targeted hypotheses from the generative algorithms but also on the quality of the *in silico* screening method.

We will introduce and review seminal works in molecular generative modeling later in this thesis in Chapter 4. However, a few pioneering works on accelerated molecular discovery that successfully completed the entire DMTA cycle should be mentioned here. In 2019, *Zhavoronkov* et al. [12] demonstrated the swift development of novel DDR1 inhibitors through a deep generative model. In only 46 days, they curated specific datasets to train generative and predictive models, carefully selected six candidates for synthesis and experimental validation, reported two compounds with nanomolar activity *in vitro* and even validated one candidate *in vivo* against mice. A study on antimicrobial discovery by *Das* et al. [13] identified 20 potential antimicrobial peptides through a generative model and molecular dynamics and synthesized and experimentally validated them in only 48

days (two peptides were found highly potent). While both these works relied on conventional, manual synthesis, *Grisoni* et al. [14] synthesized 25 AI-generated potential LXR agonists on a microfluidics platform for on-chip chemical reactions and found twelve to be potent. Major advances have also been made in material design: *MacLeod* et al. [15] created a robotic platform that can be controlled automatically via an algorithm for optimization of electronic and optic properties of thin-film materials. A work by *Yao* et al. [16] used supramolecular variational autoencoders to guide the optimization of metal-organic frameworks according to a targeted functionality such as gas separation.

1.2 Scope of the thesis

This thesis will be devoted to improve the entanglement of molecular generative models and molecular property prediction models. We believe that this interplay can become instrumental to accelerated molecular discovery as shown in the inner cycle in Figure 1.1**C**. Throughout the thesis, we will strive to build algorithms that can conditionally generate molecules for a wide range of possibly high-dimensional contextual information.

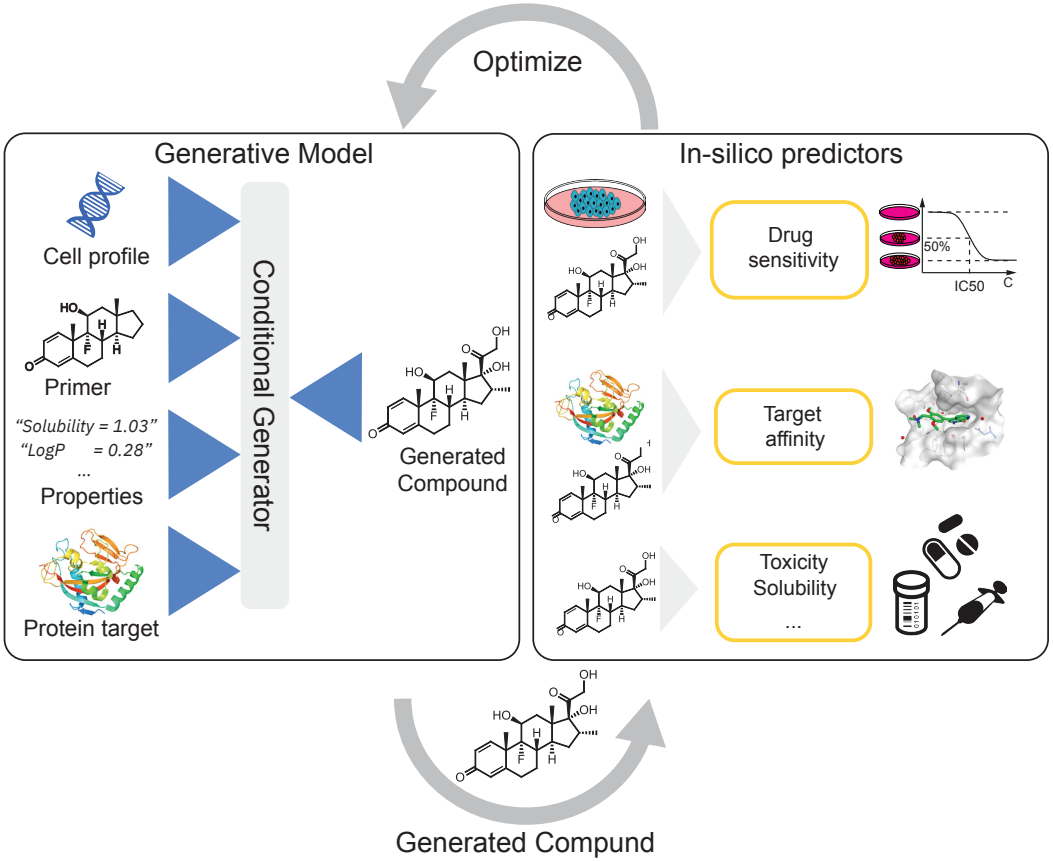
In the past years, optimizing molecular generation towards single, specific physicochemical or biochemical properties has widely been shown effective [11, 17, 18, 19, 20, 21, 22, 23, 24, 25]. However, these models suffer from two problems:

- they are task-specific and usually trained to maximize a specific property or a specific multiproperty objective. This implies that, if the objective changes only mildly, the models can not be used anymore.
- they disregard system-level information, e.g. about the cellular environment in which the drug is intended to act. Such information is notoriously challenging to integrate.

Both items point to the same underlying issue. Instead of having to tune generative models to maximize specific properties, we wish to build *generic* models that can generate molecules more flexibly. The open challenge is to build multimodal conditional generative models that can be conditioned on a wide range of desired property values (or a wide range of biological contexts) and leverage disparate sources of knowledge when generating molecules. Ideally, such models can be queried with a "semantic context" and do not require finetuning to propose targeted hypotheses. This is a substantially harder problem that has received significantly less attention. In medicinal chemistry, typical forms of the "semantic context" include e.g., (1) target proteins, (2) cell expression signatures or (3) an existing drug. If we strive to build a generative model that can be conditioned on a wide range of context, we also need to be able to evaluate any generated molecule on any possible context. In all three aforementioned cases, these evaluations are multimodal themselves as they are based on the interaction of the generated molecule, namely

(1) the binding affinity to a protein target, (2) the induced IC50 for a tumour type, or (3) a synergistic effect in polypharmacy.

A graphical abstract of the work covered in this thesis is given in Figure 1.2. As shown on the right, the first part will be devoted to developing algorithms for virtual screening.



[a] https://github.com/GT4SD

Figure 1.2: Framework for conditional molecular design with chemical language models. First, a series of predictive models for drug target binding affinity, toxicity and other molecular properties will be developed (right box). Then, generative models that can be conditioned on different, biomolecular context (cancer cell profiles, proteins, chemical scaffolds and continuous properties) will be developed (left box). Some of these models can be steered with the previously developed property predictors to sample molecules adhering to specific conditions. We then present a case-study on closed-loop molecular discovery that includes automatic retrosynthesis modeling and wet-lab synthesis. In the last chapter we propose a multitask model, the Regression Transformer, which demonstrates how predictive and generative tasks can be unified within the same model. All molecular generative models developed in this thesis are available in the GT4SD, Generative Toolkit for Scientific Discovery [26][a].

This will encompass molecular property prediction models for biochemical or physicochemical properties (e.g., toxicity or water solubility) but also proteochemometric models for predicting interaction effects between molecules and protein targets.

As visualized in Figure 1.2 (*left*), in the second part of this thesis we will formulate and apply conditional molecular generative models for four types of contexts:

1. **Cell profiles**: In this task, the objective is to develop a model that can be conditioned on an omic profile (e.g., gene expression data from a malignant tumour) and generate molecules that are likely to exhibit a high inhibitory effect against the cell profile. This model will be steered with the drug sensitivity prediction model from *Manica*, *Oskooei*, *Born*, et al. [27] that was previously developed in our team and will not be detailed herein.

2. **Protein targets**: In this task, the objective is to develop a model that can be conditioned on a protein sequence and generate molecules (i.e., ligands) that are likely to bind to the protein. This method will be steered with the molecular property prediction models developed in the first part.

3. **Molecular substructures** (e.g., scaffolds): In this task, the objective is to develop a model that can be conditioned on a seed molecule (or even an explicit, possibly discontiguous molecular substructure) and generate molecules that are 1) similar to the seed *and* 2) exhibit desired properties. Together with **4.** this will be assessed in the last part of the thesis.

4. **Continuous properties** (e.g., a desired solubility value): In this task, the objective is to develop a model that can be conditioned on a desired floating-point property value and generate molecules adhering to the property constraint. Together with **3.** this will be assessed in the last part of the thesis.

The common theme throughout the conditional molecular generative models that are devised and studied in this thesis, is that they all aim for flexibility *at inference time*. The few related works that has been devoted to this substantially harder problems will be reviewed in the respective, topical sections. Notably, throughout the entire thesis we will almost exclusively rely on *language* models. While originally developed for NLP applications, these models have recently enjoyed tremendous success in bio- and chemoinformatics [28, 29, 30].

1.3 Organisation of the thesis

This thesis is organised in two parts. In Part I we develop molecular property prediction models. This part comprises two chapters. First, in Chapter 2 we benchmark various

molecular representations and then propose and study *ToxSmi*, an interpretable chemical language model for molecular property prediction. In Chapter 3 we develop novel proteochemometric language models for protein-ligand binding affinity prediction and show that their performance can be significantly improved when relying only on active site residues rather than full protein sequences.

The core part of this thesis, presented in Part II, is devoted to building and evaluating conditional molecular generative models. In Chapter 4 we first formulate a hybrid Variational Autoencoder (VAE) that can be conditioned on a biomolecular context vector (e.g., an omic expression signature or a protein target) and generate molecules with high affinity toward this context. Critically, this model relies on reinforcement-learning optimization that is steered with the property predictors developed in Part I of this thesis. We apply this model to conditional design of SARS-CoV-2-related protein targets and present a case study on accelerated molecular discovery that integrates our generative model into a completely autonomous workflow spanning retrosynthesis modeling, synthesis protocol generation and the successful wet-lab synthesis on a robotic hardware.

In Chapter 5, the last chapter of this thesis, we propose the *Regression Transformer*, a multitask language model that abstracts regression as a conditional sequence modeling task and unifies our previous efforts on property prediction and conditional molecular generation within the same model.

1.3.1 Contributions

This dissertation is largely based on the seven publications shown below. Below each publication, a detailed author contributions statement is given. All co-authors were informed about these statements and no co-author disagreed. The publications are ordered roughly by chapter and the star * denotes shared first-authorship:

1. **J. Born***, G. Markert*, N. Janakarajan, T.B. Kimber, A. Volkamer, M. Rodriguez Martinez, M. Manica
"Chemical Representation Learning for Toxicity Prediction." *Digital Discovery* (In Review)

   Presented in Chapter 2. **JB** and MM conceived the study and the ToxSmi model which **JB** implemented. GM and **JB** conceived and implemented the SMILES transformations, implemented the remaining models and ran the experiments on representation comparison. GM, **JB** and MM conducted the attention analysis. NJ and **JB** conducted further model benchmarking. **JB** conceived and conducted the uncertainty analysis. **JB**, TBK and AV conceived the validation study which TBK conducted with **JB**'s code. AV, NJ and **JB** conducted the interpretability comparison. **JB** led the manuscript writing to which all authors significantly contributed.

2. **J. Born**, T. Huynh, A. Stroobants, W.D. Cornell, M. Manica
"Active Site Sequence Representations of Human Kinases Outperform Full Sequence Representations for Affinity Prediction and Inhibitor Generation: 3D Effects in a 1D Model." *Journal of Chemical Information and Modeling* (2022), 62, 2.

   Presented in Chapter 3 (Section 3.1 to Section 3.6) and Chapter 4 (Section 4.6). MM, **JB** and WDC conceived the study. **JB** and MM conceived the model which **JB** implemented. TH and WDC performed the sequence alignment. AS implemented the initial code for the Gaussian Process. **JB** performed and analyzed all experiments. MM and TH produced the graphical abstract and the 3D protein visualization. **JB** led the manuscript writing to which all authors significantly contributed.

3. **J. Born**, Y. Shoshan, T. Huynh, W.D. Cornell, E.J. Martin, M. Manica
"On the Choice of Active Site Sequences for Kinase-Ligand Affinity Prediction." *Journal of Chemical Information and Modeling* (2022), 62, 18.

   Presented in Chapter 3 (Section 3.7 to Section 3.8). EJM, **JB**, WDC and MM conceived the study. EJM proposed the Martin and **JB** the Combined active site definition. TH and WDC performed the sequence alignment. YS conceived and implemented the augmentation strategies and performed the experiments. **JB** implemented the models, performed all other experiments, analyzed all results and created all visualizations. **JB** led the manuscript writing to which all authors significantly contributed.

4. **J. Born**[*], M. Manica[*], A. Oskooei, J. Cadow, G. Markert, M. Rodriguez Martinez
"PaccMann^{RL}: De-novo Generation of Hit-like Anticancer Molecules from Transcriptomic Data via Reinforcement Learning." *iScience*, (2021), 24, 4.

   Presented in Chapter 4 (Section 4.2 to Section 4.3). **JB**, MM and AO conceived the study and the PaccMann^{RL} model. **JB**, MM, AO, JC and GM implemented the different components. MM preprocessed the data. **JB** conducted all experiments and analyzed all results. **JB** and MM created the visualizations. GM performed the case study on multiproperty optimization. **JB** led the manuscript writing to which all authors significantly contributed.

5. **J. Born**[*], M. Manica[*], J. Cadow[*], G. Markert, N.A. Mill, M. Filipavicius, N. Janakarajan, A. Cardinale, T. Laino and M.R. Martinez
"Data-driven molecular design for discovery and synthesis of novel ligands: a case study on SARS-CoV-2." *Machine Learning: Science & Technology* (2021), 2, 2.

   Presented in Chapter 4 (Section 4.2 and Section 4.4). **JB** and MM conceived the study. MM led the model development by **JB**, JC and GM. **JB** developed the con-

ditional generator. **JB** and JC analyzed the experimental results. **JB**, MM, JC and NAM created the visualizations. MM performed the synthesis planning experiments. MM, AC and TL selected the molecule for synthesis and oversaw the robot. **JB** led the manuscript writing to which all authors significantly contributed.

6. **J. Born**, M. Manica
"Regression Transformer: Concurrent sequence regression and generation for molecular language modeling." *Nature Machine Intelligence* (In Review)

Presented in Chapter 5. MM and **JB** conceived and designed the study together. MM conceived the tokenization scheme. **JB** conceived and implemented the objective functions, the alternating training regime and the numerical encodings. **JB** performed all experiments and analyzed all results and created all visualizations with support from MM. **JB** wrote the paper with some inputs from MM.

7. **J. Born** M. Manica
"Trends in Deep Learning for Property-driven Drug Design." *Current Medicinal Chemistry* (2021), 28, 38.

Review paper, content used throughout the thesis (mostly Chapter 1 and Section 4.1). MM and **JB** conceived this review paper, developed the software for publication keyword searches and **JB** performed and analyzed the experiments. **JB** wrote the paper with some inputs from MM.

Additional contributions.

In addition to the works listed above, the author also contributed to the following publications. Note that this lists only publications related to the content of this thesis. The list is ordered by depth of contribution:

1) J. Cadow*, **J. Born***, M. Manica*, A. Oskooei, and M. Rodríguez Martínez
"PaccMann: a web service for interpretable anticancer compound sensitivity prediction." *Nucleic acids research* (2020), 48(W1), W502-W508.

2) M. Manica, **J. Born**, J. Cadow, D. Christofidellis, A. Dave, D. Clarke, Y.G. Nana Teukam, S.C. Hoffman, M.Buchan, V. Chenthamarakshan, T.Donovan, H.H. Hsu, F. Zipoli, O. Schilter, G. Giannone, A. Kishimoto, L. Hamada, I. Padhi, K. Wehden, L. McHugh, A. Khrabrov, P. Das, S. Takeda, J.R. Smith
"GT4SD: Generative Toolkit for Scientific Discovery". *arXiv preprint arXiv:2207.03928* (2022).

3) N. Janakarajan, **J. Born**, M. Manica
"A Fully Differentiable Set Autoencoder". *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2022), 3061-3071.

4) A. Weber, **J. Born**, and M. Rodriguez Martínez.
"TITAN: T-cell receptor specificity prediction with bimodal attention networks" *Bioinformatics* (2021), 37, i237-i244.

5) N. Park, M. Manica, **J. Born,** D. Y. Zubarev, N. A. Mil, J. L. Hedrick, P. L. Arrechea, T. Erdmann
"An extensible software platform for accelerating polymer discovery through informatics and artificial intelligence development." *Nature Communications* (In Revision).

6) G.A. Tadesse, **J. Born**, C. Cintas, M. Manica and K. Weldemariam.
"MPEGO: A toolkit for multi-level performance evaluation of generative models for material discovery domains." *KDD Workshop on Machine Learning for Materials Science* (2022).

7) V. Chenthamarakshan, P. Das, S. Hoffman, H. Strobelt, I. Padhi, K.W. Lim, B. Hoover, M. Manica, **J. Born**, T. Laino and A. Mojsilovic.
"CogMol: target-specific and selective drug design for COVID-19 using deep generative models." *Advances in Neural Information Processing Systems*, 33 (2020).

Part I

Property prediction with language models

2 Molecular property prediction

2.1 Introduction

A first research question we will address is to find an ideal combination of model and data representation for molecular property prediction tasks. This will later become useful when building generative models that are driven by property predictors.
Among the typically investigated molecular properties, toxicity is particularly important in the pharmaceutical industry since it accounts for the failure of $> 30\%$ of all clinical trials [31]. A commonly utilized approach in lead compound design is to avoid molecules with toxicophores, i.e., substructures or chemical motifs that are likely to exert toxic effects [32]. Empirically, this heuristic has proven to be limited – the success rates are steadily declining and oncological pharmaceuticals are particularly affected as only 3.4% of the clinical trials are successful [33].

Deep learning has promised a methodological turnaround toward data-driven approaches to combat the ever growing need for new therapeutics [10]. Consequently, a considerable body of literature developed around molecular property prediction [34, 35, 36, 37, 38] with several works focusing on toxicity prediction [39, 40, 41, 42, 43, 44]. It is widely accepted that the ultimate success of QSAR modeling critically depends on selected molecular representation [45]. Traditional chemoinformatics models relied on 1D descriptors such as binary fingerprints [46]. While in 2018, the lion's share of publications on molecular property prediction utilized fingerprints, the usage of graph-based techniques soared rapidly in the last years and now surpassed fingerprints as the most popular representation type according to publication keyword matches [47].

2.1.1 The rise of chemical languages in chemoinformatics

SMILES (Simplified molecular-input line-entry system) is a molecular inline notation that is obtained by traversing the molecular graph. For example, benzene can be written as `c1=CC=CC=C1`. While SMILES was originally devised by *Weininger* [48] for data storage purposes, it has rapidly gained popularity for molecular property prediction tasks in the last years [27, 45, 49]. SMILES allows treating chemistry as a language – molecules can be seen as words and atoms and bonds as letters. Treating molecules as SMILES strings

opens the door to leverage tremendous recent progress in NLP such as autoregressive decoding methods or the Transformer [50].

SMILES will be fundamental throughout this thesis as they constitute the foundation for any chemical language model. The use of SMILES for molecular activity prediction was first proposed by *Jastrzebski* et al. [51] and quickly adopted for predicting drug-target interactions [52], chemical reaction products [53], drug sensitivity [27], toxicity [54] or to train molecular generative models [55]. SMILES strings are split (i.e., tokenized) into atomic units (i.e., tokens) that are passed as one-hot or learned embeddings to a neural network, similar to words in NLP. Notably, the multiplicity of SMILES enables data augmentation by traversing the same molecular graph in different ways. This was first reported beneficial for the performance of property prediction models by *Bjerrum* [56], but later confirmed in countless settings [38].

But despite the ubiquitous usage of SMILES, there is no universal, canonical SMILES representation (e.g. `PubChem` [57] kekulizes "canonical" SMILES, whereas `RDKit` [58] does not), even though several attempts toward unification were proposed [59]. Thanks to the use of SMILES, significant progress has been achieved in countless disciplines in chemoinformatics, including but not limited to: forward reaction prediction [28], retrosynthesis modeling [60], *de novo* molecular design [11], automated extraction of experimental procedures [61] and synthesis actions [30], reaction atom-mapping [29] or yield prediction [62].

In this chapter we systematically investigate the predictive power of different flavors of SMILES and compare it to established chemical descriptors or more complex representations such as graphs. We also propose *ToxSmi*, a novel, SMILES-based, robust and interpretable language model that is shown to exhibit excellent prediction performance across several datasets, focused on but not limited to toxicity. We show how ToxSmi's attention maps can be useful to understand the model's predictive process and find enrichment for known toxicophores even without explicit supervision. Since model safety and reliability is a critical aspect of drug modeling, we borrow two simple uncertainty estimation methods proposed in related fields, namely Monte Carlo dropout [63] and test-time-augmentation [64]. We provide quantitative evidence on how they can be used to identify probable misclassifications. Last, we validate ToxSmi on a large-scale proprietary toxicity dataset and find that it outperforms previous work while giving similar insights into revealing cytotoxic substructures.

2.2 Problem formulation

Let $\mathcal{M}$ denote the molecular space and $\mathcal{Y}$ denote the QSAR property scores. We are interested to learn a function $\Phi : \mathcal{M} \rightarrow \mathcal{Y}$ that maps a molecule to a property score. Φ is parameterized through our model and learned from a labelled dataset $\mathcal{D} = \{m_i, y_i\}_{i=1}^{N}$ where $m_i \in \mathcal{M}$ and $y_i \in \mathcal{Y}$.

2.3 Molecular representations

In the following we investigate three types of molecular representations:

1. **Molecular fingerprints**: Molecular fingerprints are binary vectors where the value at each position indicates the presence of a certain substructure. Here, we use extended connectivity fingerprints (ECFP [46]). We use ECFPs with 512 bits and a radius of 2 (ECFP4). In the DeepTox publication, *Mayr* et al. [39] used a rich set of chemical descriptors but found that it can be approximated well solely based on ECFP4. Therefore, we refrain from examining other types of fingerprints.

2. **Molecular graphs**: Each molecule is denoted by an undirected graph $G = (V, E)$ where vertices denote atoms and edges denote bonds. Vertices are labelled by their atom identity and edges by the bond valence.

3. **String-based representations**: We examine chemical languages such as SMILES [48] or SELFIES [65]. Due to their recent success, different types of string representations, in particular different SMILES flavors, are our main focus.

2.3.1 Chemical languages

Figure 2.1 explains the notation of SMILES and shows an overview of a multitude of SMILES flavors. All these strings correspond to the *same* molecule, they simply vary by their convention. For example, SMILES strings normally do not explicitly list bonds unless they are double (`=`) or triple (`#`). Hydrogen atoms are also not stated except if they are important for the stereoinformation of a tetrahedral center (e.g., `[C@H]`). The starting point is always the raw SMILES representation as read from the data source. We then experimented with a cascade of transformation as follows:

Chemical transformations refer to semantic changes in the visibility of certain properties in the string and include:

1. **Canonicalization**: Since a molecular graph traversal is ambiguous, SMILES are non-unique representations of molecules. Canonicalization ensures that every molecule is represented by exactly one string. Here, we use canonicalization as defined in `RDKit` [58]. Canonicalization bears the advantage of an increased data uniformity.

2. **Kekulization**: Aromatic moieties can either be represented explicitly or implicitly. In the explicit (kekulized) version, the aromatic π-electrons are static between every second carbon. Instead, in the canonical form, the electrons are delocalized (cf. Figure 2.1). The kekulized version is slightly longer but uses the same token to denote an atom, irrespective of its aromaticity.

3. **Removal of stereoinformation**: To uniquely identify a molecule from a SMILES, information abound the tetrahedral center or the double bond direction (E or Z) is sometimes needed. Since stereoinformation is rare, it is often discarded in affected molecules for reasons of simplicity and uniformity. We experiment separately with removing chirality and bond direction.

4. **Explicitness**: We experimented with making hydrogen atoms or single bonds (or both) explicit in the SMILES. This increased sequence length but also better distributes the frequency of tokens in the vocabulary.

Randomized transformations add a level of stochasticity that resembles a form of augmentation. They refer to non-chemical changes in the syntax or grammar.

1. **Augmentation**: Since SMILES are non-unique, their multiplicity can be used for data augmentation which provably improves performance of predictive [38, 56] and generative [67] models. Here, we use *online* augmentation which samples the graph traversal at runtime and generates the corresponding string.

2. **Shuffling**: *Liu* et al. [68] observed that randomly shuffling the position of the SMILES tokens does not significantly reduce performance in QSAR prediction tasks. This is striking because shuffling *destroys* the molecular identity and the local structure. It is not commonly applied. Like augmentation, shuffling occurred as a stochastic transformation at runtime.

Language translations are optional. The default language is SMILES and the only alternative language explored in this work is SELFIES. We note that other languages such as DeepSMILES [69] or the polymer language BigSMILES [70] exist.

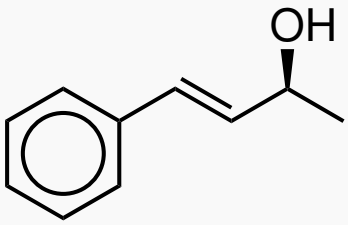
1. **SELFIES**: SELFIES is a self-referencing chemical language that overcomes the validity problem of SMILES (i.e., random SMILES strings are not generally valid) and was devised for generative models by *Krenn* et al. [65].

Tokenization splits the obtained strings into tokens. Tokens are the smallest unit of information that is presented to the model. Possible tokenization schemes are:

1. **Character-level**: Splitting on a character-level is suboptimal since it splits up atoms like Bromine (`Br`) into multiple tokens (`B` and `r`).

2. **Atom-level**: Splits strings on an atom-level to ensure that each molecular entity is represented as one feature vector to the model. By convention, this is achieved with the regular expression from *Schwaller* et al. [53].

3. **SMILES-PE (SPE)**: Inspired by byte-pair encodings [71] this method splits SMILES into substructures of varying length based on their occurrence in ChEMBL [66]. *Li* and *Fourches* [66] showed that SPE has comparable predictive to atom-based tokenization in QSAR prediction tasks. It is ideal to handle larger molecules since it drastically reduces the number of tokens.



(*S*,*E*)-4-phenylbut-3-en-2-ol

Example molecule and its different SMILES representations:

Raw: SMILES as read from data source: `c1ccc(/C=C/[C@H](C)O)cc1`

Chemical transformations

Canonical (by `RDKit`): `C[C@H](O)/C=C/c1ccccc1`
Isomeric (by `PubChem`): `C[C@@H](/C=C/C1=CC=CC=C1)O`
Without stereoinformation: `c1ccc(C=CC(C)O)cc1`
Without chirality: `c1ccc(/C=C/C(C)O)cc1`
Remove doublebond direction: `c1ccc(C=C[C@H](C)O)cc1`
Kekulization: `C1=CC=C(/C=C/[C@H](C)O)C=C1`
Explicit bonds: `c1:c:c:c(/C=C/[C@H](-C)-O):c:c1`
Explicit hydrogens: `[cH]1[cH][cH][c](/[CH]=[CH]/[C@H]([CH3])[OH])[cH]...`

Randomized transformations (non-unique!)

Multiplicity (aka augmentation) – e.g.: `C[C@H](O)C=Cc1ccccc1`
Character shuffling – e.g.: `c[C@H]Ccc/C(Cc=Oc)1/)c(`

Language translation

SELFIES: `[C][C@H1][Branch1][C][O][C][=C][C][=C][C][=C][C][=C][Ring1]...`

Tokenization

If SMILES:

Character level: `C [ C @ H ] ( O ) C = C c 1 c c c c c 1`
Atom-level [53]: `C [C@H] ( O ) C = C c 1 c c c c c 1`
SMILES Pair Encoding (SPE) [66]: `C[C@H](O) /C=C/ c1ccccc1`

If SELFIES:

SELFIES tokenizer: `[C] [C@H1] [Branch1] [C] [O] [C] [=C] [C] [=C]...`

Figure 2.1: Workflow for different chemical language flavors, showcased on (*S*,*E*)-4-phenylbut-3-en-2-ol. Translating a molecule into a string representation. Transformations can be divided into four groups and are executed sequentially. First, *chemical transformations* change the semantics of the language. *Randomized transformation* add a level of stochasticity. Then, *language transformations* optionally convert the SMILES to another language (e.g., SELFIES). Last, the *tokenization* scheme determines the feature blocks provided to the model.

For example, in our experiments, we investigated SELFIES *with* and *without* augmentation. All sequences are enclosed by a <START> and <STOP> token and are left-padded to the longest sequence in the dataset, respectively. The entire SMILES processing pipeline is implemented in the publicly available package `pytoda`[1] [72].

2.4 Model definitions

2.4.1 Language models

2.4.1.1 ToxSmi

The *ToxSmi* model is an attention-based multiscale convolutional neural network. It is inspired by a bimodal variant of this model, called `PaccMann`, which was originally developed for drug sensitivity prediction [27, 73]. The network architecture is shown in Figure 2.2. In the canonical *ToxSmi* model, each token is represented as a learned embedding of dimensionality $H = 256$. The input matrix $\mathbf{X} \in \mathbb{R}^{T \times H}$ where T is the sequence length (i.e., padding size). The embeddings $\mathbf{X}$ are processed by three parallel 1D-convolutional layers with kernel sizes 3, 5 and 11. A fourth channel has a residual connection without convolutions (not shown in Figure 2.2). For each of the four channels, we utilize a stack of $c = 6$ attention heads. In each head, a self-attention mechanism (similar to the one proposed by *Bahdanau* et al. [74]) enables the model to focus on relevant parts of the molecule. In each head, the attention weight α_i of token i is computed by:

$$\alpha_i = \frac{\exp\left(u_i\right)}{\sum_j^T \exp\left(u_j\right)} \quad \text{, where} \quad \vec{\mathbf{u}} = (\mathbf{M}\mathbf{W}_1)\vec{\mathbf{v}} \tag{2.1}$$

where $\mathbf{M} \in \mathbb{R}^{T \times C}$ is the output of the convolutional layer with $C = 128$ filters, $\mathbf{W}_1 \in \mathbb{R}^{C \times S}$ and $\vec{\mathbf{v}} \in \mathbb{R}^A$ are learnable parameters, and $S = 256$ is the dimensionality of the attention space.

For notational purposes, consider $\mathbf{A} = [\vec{\boldsymbol{\alpha}}, ..., \vec{\boldsymbol{\alpha}}] \in \mathbb{R}^{T \times C}$ as the attention matrix with the attention vector repeated C times [2]. Then, the output vector $\vec{\mathbf{e}} \in \mathbb{R}^C$ of each attention head is obtained by filtering $\mathbf{M}$ with the attention matrix (i.e., "we attend"):

$$\vec{\mathbf{x}}_{out} = \mathbf{1}^T[\mathbf{M} \circ \mathbf{A}] \tag{2.2}$$

Basically, we filter the output sequence from a given convolutional kernel with the attention scores and then we sum over the sequence dimension to obtain a single score for each filter. This is similar to the so-called Bahdanau attention from language translation [74], with the difference that 1) we refrain from having an additional $\tanh$ non-linearity (it did

[1] https://pypi.org/project/pytoda

[2] In practice, we exploit automatic tensor broadcasting.

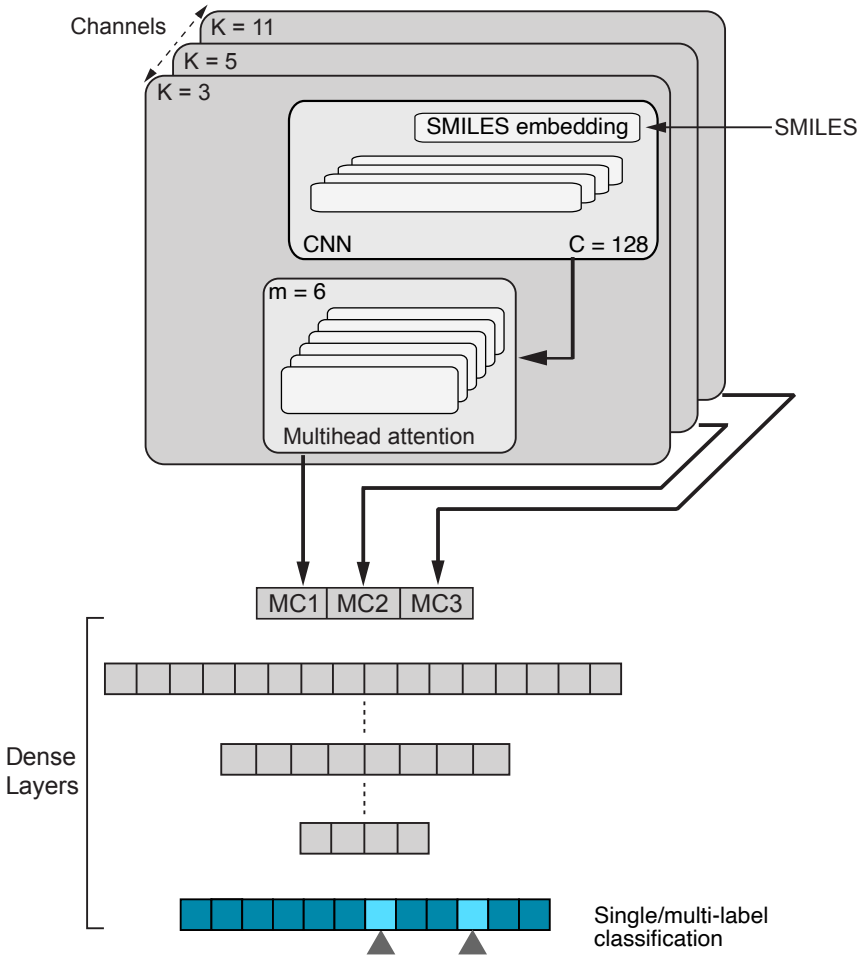


Figure 2.2: The ToxSmi model, a convolutional, attention-based neural network for molecular property prediction. The first step of `ToxSmi` is the sequence embedding. The embeddings are then used for three (parallel) 1D convolutional layers to aggregate local information. Next, the multihead self-attention mechanism calculates the attention weights and filters the inputs accordingly. The resulting outputs are concatenated and processed by a set of dense layers, resulting in one (or multiple) output scores.

not perform well in initial experiments) and 2) there is no need for additive attention because there is no output tokens to attend to. Since the attention is thus done with a single sequence we call it **self-attention**.

With three parallel convolutional layers (plus one residual connection), each with $m = 6$ attention heads, we obtain 24 output vectors $\vec{x}_{out}$ which are stacked to a single large vector. This vector is processed by a stack of dense layers ($[1024, 512]$ units) before a final

layer with a sigmoid activation computes the class-wise predictions. The model is trained with a binary cross-entropy for classification.

2.4.1.2 Recurrent networks

We examined two flavours of recurrent neural networks as alternative chemical language models; the Gated Recurrent Unit, GRU [75], and the neuromodulated Bistable Recurrent Cell, nBRC [76]. The GRU employs a gating mechanism to control information flow, making it suitable for handling longer sequences. The nBRC is a biologically inspired modification of the GRU that is superior in exact memorization and counting [76] which might be crucial in handling SMILES sequences due to their ring opening and closure symbols. This cell was never tested before in chemical language models. The cell can switch between a monostable and a bistable state and hold onto information for an arbitrarily long time period. For both models, we use two bidirectional layers with 256 units. The last hidden states from both directions are processed by a 3-layered dense network with 1024, 1024 and 512 hidden units respectively (50% dropout). The final scores are returned by an ensemble of 5 linear networks acting on the 512-dimensional representation. The models were optimized by Adam [77] at a constant learning rate of $1\mathrm{e}{-4}$.

2.4.2 Fingerprint-based models

For fingerprint-based models we used 512-bit ECFP fingerprints [46] with a radius of 2 (ECFP4).

k-nearest-neighbor (k-NN). As a non-parametric baseline we explored the k-NN algorithm and employed (inverted) Tanimoto similarity [78] as distance function. Note that the Tanimoto similarity is the same as the Jaccard index. We set $k = 23$, based on the performance on the Tox21 *test* dataset.

Dense neural network (DNN). This was a simple, four-layered, fully-connected neural network with $[512, 1024, 2048, 1024]$ units and a sigmoid activation function was used.

2.4.3 Graph-based models

Molecular graph representations were examined with graph neural networks and graph kernels.

Graph convolutional network

Graph convolutional networks (GCN) are GNNs which use convolutions as neighborhood aggregation function. Following [79], this is a GCN with two graph-convolutional

layers (64 units) and one dense layer (128 units), no dropout, 75 atom features and a sigmoid activation function.

Graph kernels

Graph kernels rely on a kernel $k(x, x')$ that measures similarity between molecular graphs x and x' [80]. We experimented with four different kernels.

1. **Shortest-path (SP)**: This path kernel [81] first transforms the graphs G_1, G_2 into shortest-path graphs S_1, S_2 using the Floyd algorithm [82]. Let $S_1 = (V_1, E_1)$ and $S_2 = (V_2, E_2)$, then our shortest-path kernel is:

$$k_{shortest-paths}(S_1, S_2) = \sum_{e_1 \in E_1} \sum_{e_2 \in E_2} k^{(1)}_{walk}(e_1, e_2) \tag{2.3}$$

   where $k^{(i)}$ walk is a positive definite kernel on edge walks of length 1.

2. **Weisfeiler-Lehman (WL)**: This subtree kernel relies on the Weisfeiler-Lehman (WL) relabeling method [83]. Let $G_n = (V, E, l_n)$ and $G'_n = (V', E', l'_n)$ be the n-th iteration rewriting of the graphs G and G'. Then the WL kernel is defined as

$$k^h_{WL}(G, G') = \sum_{n=0}^{h} k_\delta(G_n, G'_n) \tag{2.4}$$

$$\text{where } k_\delta((V, E, l), (V', E', l')) = \sum_{v=V} \sum_{v' \in V'} \delta(l(v), l'(v')) \tag{2.5}$$

   where δ is the Dirac kernel.

3. **Message Passing (MP)**: This subtree kernel [84] extends the concept of message-passing [85] from GNNs to graph kernels. It's a generalization of the WL kernel that uses a smoother definition of structural equivalence.

4. **Wasserstein-Weisfeiler Lehmen (WWL):** This extension of the WL kernel relies on the Wasserstein distance between node feature vector distributions of the WL subgraphs. For details see *Togninalli* et al. [86].

In all cases, the graph kernels were used to measure sample similarity and a SVM [87] was employed for classification.

2.5 Studied datasets

2.5.1 Tox21 Dataset

The Tox21 dataset [88] includes $12,707$ compound entries of small molecules tested on 12 targets [89], classified as toxic or non-toxic. Five of the 12 targets are associated with hormones (such as the estrogen receptor, the androgen receptor and aromatase). The dataset comes with a fixed split of $11,764$ *training*, 296 *test* and 647 *score* molecules. *Test* labels were withheld from participants during the original challenge and used to compute the leaderboard, but later made available so that participants could refine their models for the final evaluation on *score*.

2.5.2 MoleculeNet datasets

The MoleculeNet benchmark [34] distributes a variety of datasets from quantum mechanics over physical chemistry and biophysics to physiology.

1. **BACE**: This is a dataset of 1522 inhibitors against human β-secretase 1 (BACE-1), represented quantitatively by their IC_{50} value and qualitatively through binary labels indicating their inhibition success [90]. The 2-D structures of these compounds and IC_{50} values are gathered from experimental scientific literature. Scaffold splitting is recommended for this dataset.

2. **SIDER**: Side Effect Resource (SIDER) is a database of 1427 approved drugs and their associated adverse drug reactions (ADR), grouped into 27 system organ classes [91, 92].

3. **ClinTox**: This dataset comprises 1491 compounds and compares drugs approved by the FDA with those that have failed clinical trials due to toxicity through 2 classification tasks - clinical trial toxicity status and FDA drug approval status [93, 94].

4. **BBBP**: The Blood–brain barrier penetration (BBBP) dataset contains information on the blood-brain barrier permeability properties of over 2000 compounds [95]. This is a classification task where binary labels are provided and a scaffold split is recommended.

5. **HIV**: The HIV dataset contains information about the ability of $40,000$ compounds to inhibit HIV replication [96]. The classification task categorizes compounds as being active or inactive against HIV replication. A scaffold split is recommended for this dataset.

6. **Tox21**: This is a redistribution of the original Tox21 dataset [88]. This distribution is much smaller (8014 molecules), SMILES are largely canonicalized and it does

not come with a fixed split. A random split is recommended. The ToxSmi model was trained on this dataset will be used later for generative tasks and will be denoted by θ_{Tox21}.

On each dataset we trained ten models on repeated data splits in the recommended strategy. For the comparison to Grover [36], who trained the models on scaffold splits for *all* datasets, we trained additionally ten models on scaffold splits for the affected datasets (SIDER, ClinTox, Tox21).

2.5.3 Cytotoxicity dataset

As an external validation, a cytotoxicity dataset compiled by the Leibniz-Forschungsinstitut für Molekulare Pharmakologie (FMP) was employed [97]. The data collected by the FMP measures the cytotoxicity of molecules and was initially used in the study by [98]. The relative growth of two cell lines, namely HEK292 (kidney) and HepG2 (liver), is measured. A molecule is considered cytotoxic if it inhibits growth by at least 50% on one or both of the two cell lines. Before pre-processing, the data set before consists of $34,848$ measured compounds. Pre-processing of the data is done in the same manner as in the original study [98]. More specifically, it uses `RDKit` and consists of a sanitization, a standardization, and a de-duplication step, resulting in $34,366$ compounds. Out of these molecules, only 4.65% are labeled cytotoxic, leading to a highly imbalanced, yet consistent, data set. The experiments with this data set were run using high-performance computer (HPC) services from the Freie Universität Berlin [99]. To compare to the feed-forward neural network (FNN) by *Webel* et al. [98], we used a 10-fold stratified cross validation split with 10% held-out data for testing, like done in their work. Note that *Webel* et al. [98] used 2048-bit ECFP4 fingerprints.

2.6 Evaluation procedure

All models were evaluated on performance metrics that are in alignment with previous work on those datasets [34, 88, 98]. The main metric is area under the ROC curve (ROC-AUC). For the cytotoxicity dataset we report the true positive rate (TPR, also called sensitivity), the true negative rate (TNR, also called specificity) and the balanced accuracy ($\frac{TPR+TNR}{2}$).

All models were trained for 200 epochs with early stopping, the ADAM optimizer [77] and a cross-entropy loss. Learning rate varied across models, but unless otherwise specified was set to $1e{-}4$. Emulating the original Tox21 challenge, the hyperparameters of the models were tuned using the *test* dataset using raw SMILES as inputs. After the optimal configuration was found, 10 models were trained for each investigated dataset. For the original Tox21 dataset, the 10 models were obtained from identical training data with different weight initialization. For the remaining datasets, the random split was repeated for

each run. Note that we refrained from further optimizing any hyperparameters on any of the remaining datasets.

2.7 Performance results

2.7.1 Comparing molecular representations (Tox21)

In Table 2.1, we display the performance of all algorithms as measured by ROC-AUC. The performances refer to the Tox21 *score* dataset which was used to determine the Tox21 challenge winners. Comparing the model classes shows that graph kernels generally yielded the worst performance. Since the complexity of graph kernels scales quadratically with the size of the dataset, reports on graph kernels on datasets with $> 10,000$ examples are scarce to absent [100]. They are predominantly useful in small data regimes.

The SMILES representations used to train the ToxSmi model (cf. Table 2.1a) yielded performance that statistically significantly surpassed the DNN trained on ECFP as well as all graph kernel techniques in all cases ($p < 0.0.5$, one-sided Mann-Whitney-U test, U). Comparing the SMILES representations, the best results were not obtained by consistently formatting the SMILES, but rather by SMILES augmentation [56], which resembles a form of data augmentation by exploiting the multiplicity of SMILES for each molecule. This SMILES augmentation model outperforms all other models ($p < 1\mathrm{e}{-4}$, U) that do not use augmentation. This is in alignment with prior work reporting superiority of SMILES augmentation to canonical SMILES [38, 56]. Generally, the differences between the different SMILES representations were minor. Stereochemistry information stemming from chirality tokens (`/` and `\`) or bond direction (`[C@H]`) tend to confuse the model as removing them yielded slightly better performance, maybe explainable by their scarcity in the training data (18% and 7% respectively). Notably, even though SELFIES were devised for generative tasks [65], they barely rank behind SMILES regarding their predictive power for toxicity prediction. Moreover, SELFIES benefits from augmentation just like SMILES does. While, overall, the semantic (i.e., chemical) transformations of SMILES have minor impact on performance, the language transformations and tokenization scheme can be critical. For example, SMILES pair-encoding [66] gave much worse performance than all other SMILES representations, maybe because the sequences are shorter and the vocabulary is much larger leading to sparsity. We hypothesize that more labelled data or pretraining on SMILES-PE could have closed this gap. A staggering finding is that shuffling the SMILES did not change the performance significantly. This result is in accordance to *Liu* et al. [68] on other datasets and suggests that instead of aggregating local information in the SMILES sequences, the models predominantly make predictions similar to a bag-of-words model. If structural information is stripped off, the models can only rely on atom counts. However, the shuffling can be interpreted as

Representation	ROC-AUC
raw SMILES	$0.832^{*}_{\pm 0.005}$
canonical SMILES	$0.830^{*}_{\pm 0.008}$
kekulized SMILES	$0.830^{*}_{\pm 0.006}$
aug. SMILES	$\mathbf{0.853}_{\pm 0.003}$
SMILES w/o chirality	$0.834^{*}_{\pm 0.004}$
SMILES w/o bond direction	$0.834^{*}_{\pm 0.006}$
SMILES w/o bond direction & chirality	$0.835^{*}_{\pm 0.006}$
Kekulized w/o bond direction & chirality	$0.831^{*}_{\pm 0.004}$
SMILES with explicit bonds	$0.834^{*}_{\pm 0.003}$
SMILES with explicit hydrogen	$0.829^{*}_{\pm 0.007}$
SELFIES	$0.827^{*}_{\pm 0.007}$
aug. SELFIES	$0.852_{\pm 0.004}$
shuffled SMILES	$0.830^{*}_{\pm 0.003}$
SMILES Pair Encoding (PE)	$0.776^{*}_{\pm 0.01}$
aug. SMILES PE	$0.825^{*}_{\pm 0.005}$

(a) Different molecular string notations used to train the ToxSmi model.

Model	Repr.	ROC-AUC
k-NN	512-bit	0.759
DNN	ECFP4	$0.777_{\pm 0.004}$
GRU	raw	$0.781_{\pm 0.003}$
nBRC	SMILES	$0.756_{\pm 0.002}$
SP	Graph kernels	$0.567_{\pm 0.108}$
MP		$0.703_{\pm 0.040}$
WL		$0.754_{\pm 0.019}$
WWL		$0.758_{\pm 0.023}$
GCN	Graphs	$0.828_{\pm 0.008}$

(b) Remaining molecular representations and model architectures. All models were significantly inferior to the ToxSmi model with augmented SMILES ($p < 0.05$, U).

Table 2.1: ROC-AUC values on the Tox21 dataset for different algorithms and molecular representations. Each model was trained 10 times, standard deviations are shown. The best ROC-AUC values were obtained with augmented SMILES and the MCA architecture (marked in bold). Second-best performance is underlined. Models denoted with a star are significantly outperformed by the best ToxSmi model (augmented SMILES, $p < 0.05$, U). For the k-NN there were no repeated experiments.

another way of data augmentation since it is performed stochastically at runtime. In that sense, it is worth mentioning that the SMILES augmentation performed significantly better than shuffling. Last, the two RNN-based models operating on raw SMILES (GRU and nBRC) performed much worse than the ToxSmi model, suggesting the superiority of our architecture.

2.7.2 Comparing ToxSmi to prior art

The previous section revealed the superiority of the ToxSmi architecture over several baseline models. It also showed the benefit of SMILES augmentation. We therefore sought to validate the ToxSmi model on related tasks beyond toxicity prediction on the MoleculeNet benchmark [34].

This benchmark comprises several datasets about biophysics and physiology. To ensure a fair comparability, we excluded previous work whenever the data splitting strategy was not clear or no repeated experiments were conducted. Note that the Tox21 dataset listed in this section differs from the original one by *Huang* et al. [88].Since the derivative distribution by *Wu* et al. [34] is frequently used for benchmarking we retrained ToxSmi on this flavor of Tox21.
The results on all six datasets are shown in Table 2.2 and underline the superiority of our model to previous approaches, including graph convolutional networks [34] and several variants of message-passing neural networks [85], in particular the directed MPNN [35], attention-MPNN, edge-MPNN and SELU-MPNN [101]. Even the work by *Shen* et al. [102] who build a CNN based on a highly customized featurization pipeline including thirteen multidimensional descriptor classes and three fingerprint types was significantly outperformed by ToxSmi, a purely SMILES based model that did not incorporate any topological or structural features directly.

In the analysis in Table 2.2, we relied for each dataset on the data splitting strategy recommended by *Wu* et al. [34]. However, splitting molecules randomly between training and testing often results in overly optimistic model performance. This can be due to data collection biases such as, for example, sparse coverage of the chemical space or sequential decision procedures driven on previous experimental results [103]. Instead, splitting the scaffolds rather than the molecules poses a more challenging task that might approximate better the generalization performance. We therefore re-assessed the performance on three datasets where a random split was recommended (SIDER, ClinTox, Tox21). The scaffold split on those datasets enabled a fair comparison on an additional benchmark, reported by *Rong* et al. [36]. They evaluated a wide range of prediction models on scaffold-splits of all MoleculeNet datasets and then proposed GROVER, a large-scale graph Transformer that was pretrained with self-supervision on $> 10M$ molecules. The results are displayed in Figure 2.3 and include comparisons to fully-connected networks (TF_Robust [104]), three graph-convolutional networks (GraphConv [105], Weave [106] and SchNet [107]), four message-passing graph neural networks (a vanilla GNN [108], the MPNN [85] and its

Dataset Split	BACE Scaffold	SIDER Random	Clintox Random	BBBP Scaffold	Tox21 Random	HIV Scaffold	**Average**
ToxSmi (ours)	**0.861**$_{\pm 0.04}$	<u>0.659</u>$_{\pm 0.04}$	0.878$_{\pm 0.00}$	**0.915**$_{\pm 0.02}$	**0.858**$_{\pm (0.05}$	**0.813**$_{\pm 0.03}$	**0.831**
GraphConv *Wu* et al. [34]	0.783$_{\pm 0.01}$	0.638$_{\pm 0.01}$	0.807$_{\pm 0.05}$	0.690$_{\pm 0.01}$	0.829$_{\pm 0.01}$	0.763$_{\pm 0.02}$	0.752
Weave *Wu* et al. [34]	0.806$_{\pm 0.00}$	0.581$_{\pm 0.03}$	0.832$_{\pm 0.04}$	0.671$_{\pm 0.01}$	0.820$_{\pm 0.01}$	0.703$_{\pm 0.04}$	0.736
D-MPNN *Yang* et al. [35]	0.838$_{\pm 0.06}$	0.646$_{\pm 0.02}$	**0.894**$_{\pm 0.03}$	<u>0.888</u>$_{\pm 0.03}$	<u>0.845</u>$_{\pm 0.002}$	<u>0.794</u>$_{\pm 0.02}$	<u>0.818</u>
SELU-MPNN *Withnall* et al. [101]	–	0.632$_{\pm 0.01}$	–	0.693$_{\pm 0.06}$	0.820$_{\pm 0.01}$	0.747$_{\pm 0.01}$	–
AMPNN *Withnall* et al. [101]	–	0.639$_{\pm 0.01}$	–	0.709$_{\pm 0.04}$	0.812$_{\pm 0.02}$	0.742$_{\pm 0.02}$	–
EMPNN *Withnall* et al. [101]	–	0.651$_{\pm 0.01}$	–	0.705$_{\pm 0.02}$	0.829$_{\pm 0.01}$	0.759$_{\pm 0.01}$	–
MMNB *Shen* et al. [102]	<u>0.849</u>	**0.680**	<u>0.888</u>	0.739	0.842	0.777	0.796

Table 2.2: ROC-AUC values on the MoleculeNet datasets for different algorithms. With the exception of ClinTox, ToxSmi obtained always either the best (**bold**) or second-best (<u>underlined</u>) performance on each dataset. Across all datasets, ToxSmi outperforms all competing approaches. For each dataset, ten models were trained and the splitting strategy recommended by MoleculeNet was utilized.

variants D-MPNN [35] and MGCN [109]), an *n*-gram model [110], and two graph transformer networks (AttentiveFP [37] and GROVER itself [36]). The results demonstrate that ToxSmi consistently obtained superior performance compared to all flavors of fully-connected, graph-convolutional or message-passing neural networks. Only the group of graph-transformer networks (AttentiveFP and GROVER) outperformed ToxSmi; on average by 0.7% and 3% respectively. GROVER, the only model that consistently beats ToxSmi, is significantly larger and more complex. ToxSmi contains only $5M$ parameters (exact number depend on dataset/vocabulary size), consists of vanilla convolutional layers coupled with a plain Bahdanau-style attention and was trained from scratch on SMILES sequences (with augmentation). Instead, GROVER employs an order of magnitude more parameters ($50M$) and relied on large-scale pretraining on $> 11M$ molecules that utilized 250 GPUs.

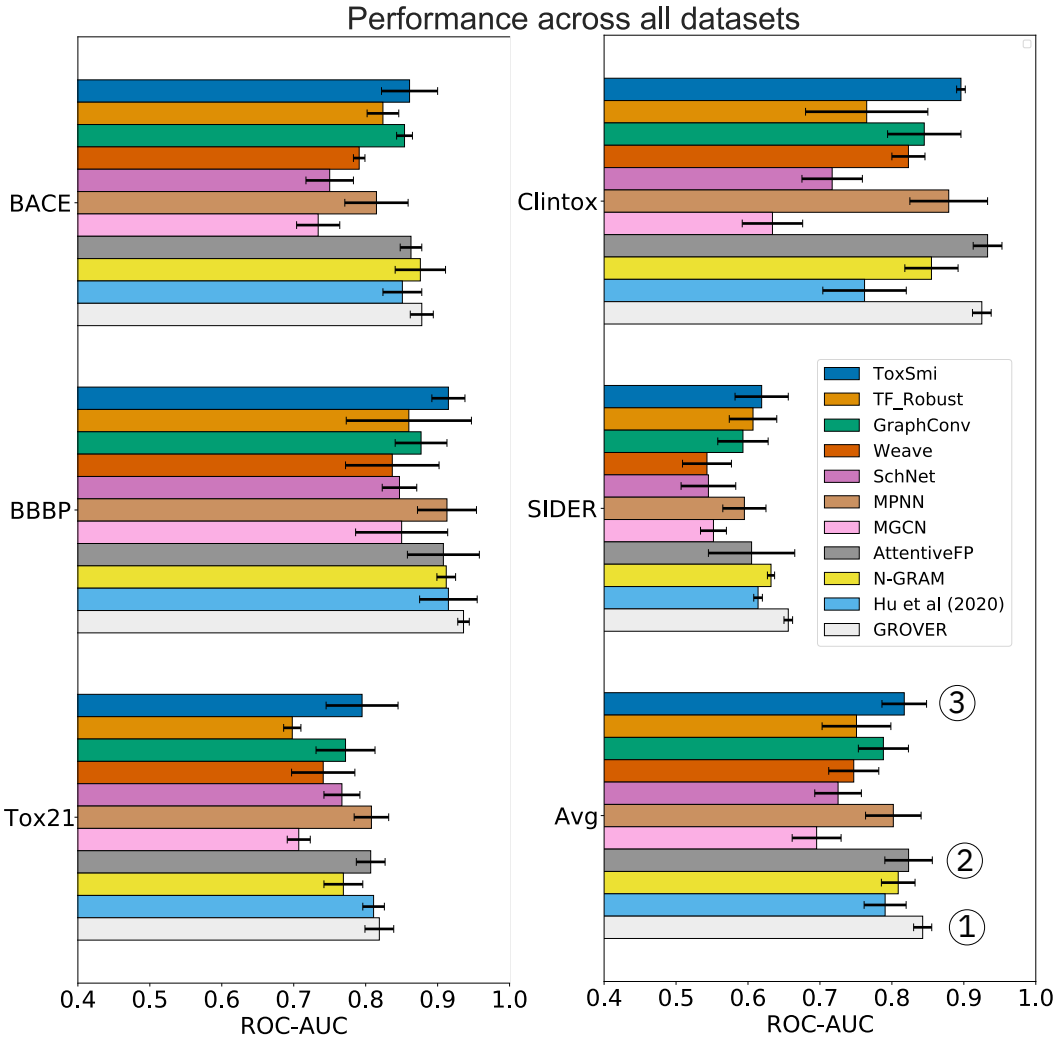


Figure 2.3: Comparison of ToxSmi to previous work exclusively on scaffold splits several MoleculeNet datasets. Overall, ToxSmi is the third best model, only surpassed by GROVER [36] and AttentiveFP [37]. For each dataset the average ROC-AUC across all tasks is reported. Results for ToxSmi were obtained by measuring test performance for 10 repeated scaffold splits. All other numbers are taken from *Rong* et al. [36] who trained all models on 3 repeated scaffold splits. The numerical results for this barplot can be found in Table A2.1.

2.8 Interpreting attention weights

Since ML models are often considered black-box by chemists, model interpretability is a heavily sought-after trait in QSAR modeling. As reviewed by *Jiménez-Luna* et al. [111], considerable previous work has been invested to explain molecular property prediction models.

In this section, we describe how ToxSmi achieves high model explainability via its self-attention mechanism. While previous methods, e.g., *Ding* et al. [112] or *Jiménez-Luna* et al. [113] rely on post-hoc workflows or (integrated) gradient schemes to become interpretable, the attention mechanism in ToxSmi is an ante-hoc method that produces attention maps as a byproduct of any prediction. We show how the attention maps can be useful to understand the model's predictive process and find that the attention maps align, in many cases, with prior knowledge in chemistry.

This experiment is done on the Tox21 dataset with ToxSmi trained on augmented SMILES sequences.

2.8.1 Analyzing molecular attention on Tox21

Extraction procedure

As can be seen in Equation 2.1 and Equation 2.2, any forward-pass through ToxSmi produces attention scores α_i, assigning relevance to each token i. We summed the scores across all heads and layers to obtain a single attention score per token. Next, we assessed whether the attention scores carry any meaning regarding the toxicity of the respective token (i.e., atom/bond). Therefore, we relied on so-called *toxicophores*, molecular substructures that are known to have toxic effects [114]. We focus on two toxicity endpoints, acute aquatic toxicity (99 alerts, see [115, 116]) and endocrine disruption (35 alerts, see [117]) that are most similar to the Tox21 tasks.

Attention on atom- or bond tokens were considered for the analysis whereas attention on other tokens (e.g., ring tokens) were discarded since it could not always be determined whether these tokens belong to a toxicophore or not. Next, the SMILES strings of all test molecules were queried against the desired toxicity alerts (given in SMARTS [118]). Whenever a match was obtained (by a substructure match in `RDKit`), the SMILES tokens affected by the alert were assigned as toxicophore tokens whereas the other tokens kept their status. For each molecule, this resulted in a grouping of the attention weights to either belonging to a toxic or non-toxic substructure.

Result

In Figure 2.4, we show the attention weights of the best ToxSmi model for all molecules from the Tox21 *score* dataset. It can be seen that, in many cases, the model assigned high attention to atoms that belong to known toxicophores. To assess whether the model selectively focused on informative substructures, we compared the mean attention weight on the toxicophoric parts of all molecules to the mean attention weight of the remaining part. Molecules that exclusively consisted of toxicophores were excluded from the analysis. This revealed a significantly higher mean for toxicophore substructures ($p = 0.011$ in two-sided Mann-Whitney-U test) showing that the model focused predominantly on toxic substructures. This is remarkable given that the attention scores were

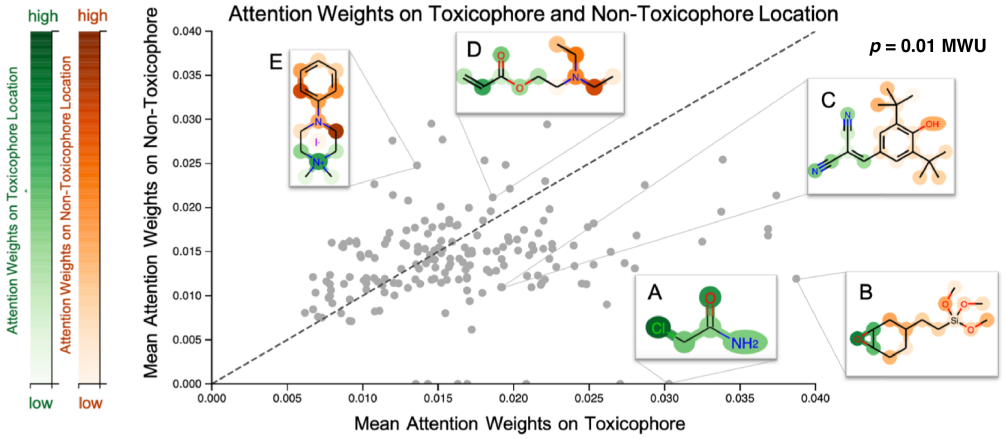


Figure 2.4: Visualization of attention maps of Tox21 compounds. Scatterplot of the mean attention of the toxicophore part of the molecule versus the mean attention of remnant. Toxicophoric atoms are colored in green, remaining parts are colored in red, intensity encodes the attention weight. Toxicophores are assigned significantly higher attention weights. Interactive visualization available at: `https://ibm.biz/tox21_attention`.

learned entirely unsupervised. While several related studies on proteochemometric modeling claimed via case studies that similar SMILES attention mechanism could automatically re-discover biochemical concepts such as protein binding sites [119, 120], *Li* et al. [121] demonstrated later in a quantitative analysis that performance was not exceeding chance level. Instead, on the Tox21 dataset, *Mayr* et al. [39] reported that the activation of a significant number of hidden neurons could be associated to toxicophore features. However, their analysis was done on training molecules and involved significant post-hoc experimentation whereas ToxSmi produces attention scores *en passant* the forward pass. Moreover, we emphasize that the attention maps are *global*, i.e., not specific for an assay and thus task-specific inference is limited to single-task classifications.

2.9 Assess trustworthiness via uncertainty estimation

Model reliability and trustworthiness are critical aspects in molecular property prediction models and have received growing attention in the past years [40, 122, 123, 124]. One realm with significant progress is the area of uncertainty estimation in neural networks [125]. To assess prediction uncertainty in toxicity models, nested cross-validation [126], snapshot ensembling [123] and conformal prediction [40] were used. Later, Monte Carlo Dropout (MC Dropout) was shown to achieve the same at a lower computational and implementational cost [124].

2.9.1 Aleatoric & epistemic uncertainty estimators

In this section, we employ two post-hoc techniques to measure prediction uncertainty. Specifically, we assess epistemic (model) uncertainty with MC Dropout, a method that draws Monte Carlo samples from the approximate predictive posterior by performing repeated forward passes of an input sample while the network's dropout layers are turned *on* [63].

Moreover, we approximate aleatoric (data) uncertainty, an uncertainty measure independent from epistemic uncertainty, via test-time data augmentation as suggested by *Ayhan* and *Berens* [64]. In our case, we perform repeated forward passes with different SMILES strings corresponding to the same molecule. From the resulting prediction ensembles, the confidence estimate c_i of sample i was obtained by scaling the sample's standard deviation to the range $[0, 1]$ and interpreting it as inverse precision:

$$c_i = -\left(\frac{\sigma_i - \sigma_{min}}{\sigma_{max} - \sigma_{min}}\right) + 1\,, \tag{2.6}$$

where σ_i is the sample standard deviation of the prediction ensemble, σ_{min} is the minimal standard deviation (0, i.e., all predictions are identical) and σ_{max} is the maximal standard deviation (0.5, i.e., 50% of the predictions are 0 and 50% are 1; we assume a binary classification setting). We further propose to use μ_i, the sample mean of the prediction ensemble, as alternative prediction and show that it yields improved performance. In practice, 200 forward passes were performed for both methods. The dropout value in the ToxSmi model was 0.5.

2.9.2 Uncertainty results on Tox21

These experiments were conducted on the Tox21 dataset (MoleculeNet flavor) and results are shown in Figure 2.5. Both epistemic and aleatoric model uncertainty were computed for each sample and each of the 12 toxicity assays and subsequently converted into a confidence estimate (cf. Equation 2.6). Both confidence estimates are strongly negatively correlated with the residual of the prediction: the higher the error, the lower the confidence (Pearson's $r = -0.558$ and -0.536 for epistemic and aleatoric confidence respectively). When averaging both estimates (their correlation is ~ 0.8), we obtain a single confidence estimate that is even stronger negatively correlated (see Figure 2.5**A**). While the average confidence is with a value of 0.94 relatively high, a known phenomenon [127], comparing the mean confidences of correctly and incorrectly classified samples reveals significant relative differences (0.96 versus 0.85). In a real world scenario of screening large-scale virtual libraries, this difference could be used out-of-the-box to eliminate molecules where predictions are more likely to be incorrect. A specific example on the benefit of the confidence estimation is shown in Figure 2.5**C**. While Bisphenol E was correctly predicted as toxic for the NR-ER assay, Phenoxypropazine was incorrectly predicted toxic. The

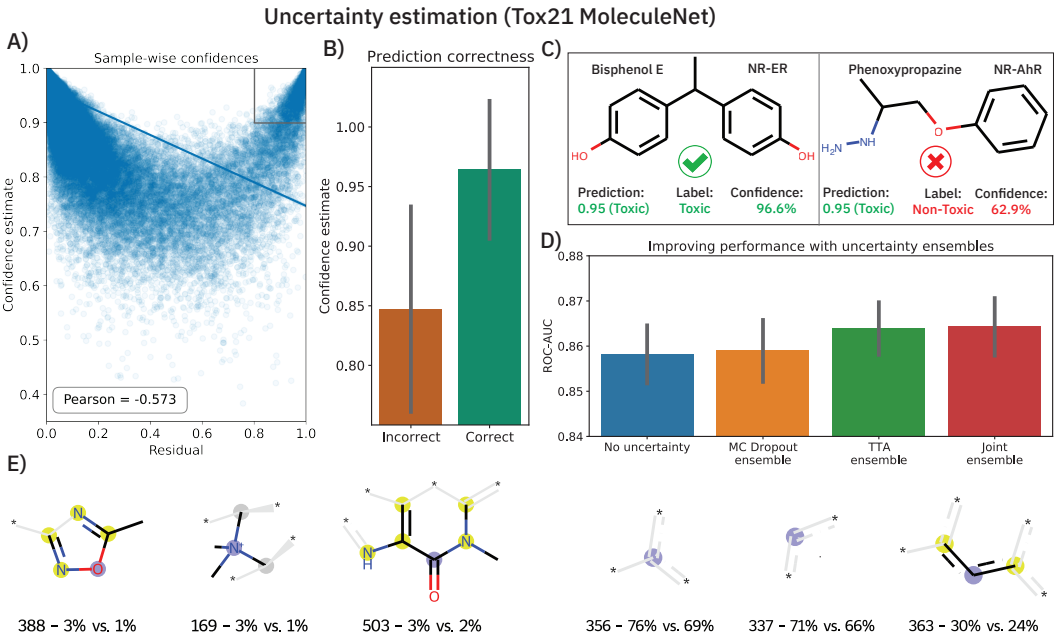


Figure 2.5: Uncertainty estimation analysis on Tox21 MoleculeNet dataset. A) Scatterplot of prediction residual and confidences reveals a strong negative correlation. **B)** Confidence estimates are significantly lower for incorrectly classified samples. **C)** Two exemplary molecules, both predicted as toxic, with an incorrect prediction identified by a low confidence estimate. **D)** The prediction ensembles formed by MC Dropout or TTA can significantly improve the prediction accuracy of the model. All plots show results across all 10 splits. **E)** The six fragments that were found most predominantly in incorrectly-classified high-confidence (ICHC) molecules (see *A)*, gray box). For each fragment, we display the ECFP4 bit and the percentage of ICHC molecules and remaining molecules where this fragment was present.

model's internal class probabilities are 0.95 in both cases (1 means toxic and 0 non-toxic) and thus do not allow to draw conclusions[3]. However, investigating the respective prediction confidences can reveal that Bisphenol E was a true positive while Phenoxypropazine was a false positive.

The scatterplot in Figure 2.5**A** reveals a small subset of incorrectly-classified high-confidence (ICHC) molecules (see gray box). These incorrect predictions are particularly undesired as they cannot be recognized and removed with our method. We inspected the molecules in the gray box (confidence $>$ 0.9, residual $>$ 0.8) more closely, aiming to identify fragments that occur commonly in ICHC molecules but rarely in the remaining molecules. In Figure 2.5**E**, we show the six ECFP4 bits that were most indicative for ICHC molecules. In a real-world scenario, such an analysis

[3] even though class probabilities are generally insufficient confidence estimators [63], they are frequently misused in practice for this task

could easily increase robustness since molecules that include these bits could be removed from the screening library. The three bits shown in Figure 2.5**E** (*left*) had the largest difference in relative occurrence between ICHC and remaining molecules, whereas the three bits shown on the right had the largest difference in absolute occurrence. Some of these fragments can be linked to tremendous recent literature, for example bit 388 corresponds to a *1,2,4-oxadiazole* ring. *1,2,4-oxadiazole*-derivatives have been largely neglected by medicinal chemistry until 2005, but in the past 15 years, research grew exponentially [128] and only in 2022 researchers reported novel cytotoxic [129], fungicidal [130], anti-inflammatory [129], antiparasitary and antiproliferative [131] effects of *1,2,4-oxadiazole* derivatives.

2.9.3 Uncertainty estimators form ensembles

The benefits of using MC Dropout and TTA are, however, not limited to confidence estimation. The prediction ensembles formed by both methods can further be used to improve the predictions. As demonstrated in Figure 2.5**D**, replacing the baseline predictions (blue), with the mean of the 200 predictions obtained from MC Dropout or TTA, improves the ROC-AUC on the Tox21 MoleculeNet benchmark from 0.858 ± 0.001 to 0.859 ± 0.001 (MC Dropout) and 0.864 ± 0.001 (TTA). Last, a late-fusion average of both techniques yields the best performance (0.865 ± 0.001) which is significantly superior to the baseline model across 10 splits ($p < 0.01, W+$).

2.10 Validation on proprietary cytotoxicity dataset

In a case study, we validated the performance of the ToxSmi model on an external data set from the FMP [97].

FMP dataset description

This dataset is comparably large ($> 34,000$ molecules) and indicates for each molecule whether it inhibited relative growth in a kidney or a liver cell line by at least 50% (for details see Subsection 2.5.3). Since this dataset is not generally available to the public, it can serve as an ideal tool to validate our method for potential proprietary use.

Performance comparison

An in-depth study on this highly imbalanced cytotoxicity dataset has been performed by *Webel* et al. [98]. The comparison of their FNN (a fully-connected network trained on

ECFP4s) to our ToxSmi model is shown in Table 2.3. Both models achieve good per-

Model	Source	Bal. Acc.	TPR	TNR
FNN	*Webel* et al. [98]	$68.89_{\pm 1.46}$	$61.57_{\pm 7.39}$	$76.22_{\pm 6.62}$
ToxSmi	**Ours**	$\mathbf{73.85}_{\pm 2.17}$	$\mathbf{69.81}_{\pm 5.82}$	$\mathbf{77.88}_{\pm 5.50}$

Table 2.3: Performance on cytotoxicity data. Mean and standard deviations of the test data performance are reported across a 10-fold cross validation. The best performance for each metric is highlighted in bold. TPR corresponds to sensitivity and TNR to specificity.

formances on this highly imbalanced cytotoxicity data set. The mean balanced accuracy of the FNN model is 68.89, whereas ToxSmi reaches a significantly better value (73.85). Three major factors that might have induced the better performance of the ToxSmi model are: 1) the use of SMILES sequences has been reported to be superior to Morgan fingerprints [27, 38]; 2) the use of SMILES augmentation which independently has been shown beneficial [56, 132] and 3) the more refined model architecture using an attention-mechanism combined with convolution to aggregate local information.

Toxicophore analysis

In the study by *Webel* et al. [98], 17 compounds (7 non-toxic and 10 toxic) from the dataset were selected and published for toxicophore analysis. The same 17 molecules are inves-

Alert class	# alerts	# matches
Genotoxic carcino- & mutagenecity	69	5
Acute Aquatic Toxicity	54	0
Hepatotoxicity	36	18
Idiosyncratic toxicity	32	9
Mitochondrial Toxicity (MT)	17	0
Developmental and MT	12	0
Non-genotoxic carcinogenicity	5	4
Kidney Toxicity	4	0

Table 2.4: Overview of toxic alerts: Subset of 229 alerts, originating from *Ji* et al. [40], used for the 17 compounds from the FMP data analysis.

tigated in this study regarding their attention weights. We extracted toxicophores using 229 substructures from 8 alert classes (see Table 2.4). This was a subset of the list of 3800 structural alerts from 22 alert classes from the emoltox server; kindly provided by *Ji* et al. [40]. The selection was done to better represent toxic effects more related to the cytotoxic effects measured in liver (HEPG2) and kidney (HEK292) cells. This lead to the exclusion

of unspecific alerts for e.g. extended functional groups, PAINS, and others (see final list of alert classes in Table 2.4).

From the set of 17 compounds, three case studies as described here, the first two are also discussed by *Webel* et al. [98]. While molecule 1 (Figure 2.6A) represents a false negative based on the predicted score, high attentions are attributed to the tertiary substituted ethylendiamine, a similar toxicophore as identified in the study by *Webel* et al. [98] and as caught by the known toxicophores from eMolTox [40], pointing to genotoxic carcinogenicity, mutagenicity and hepatoxicity. The second molecule was correctly pre-

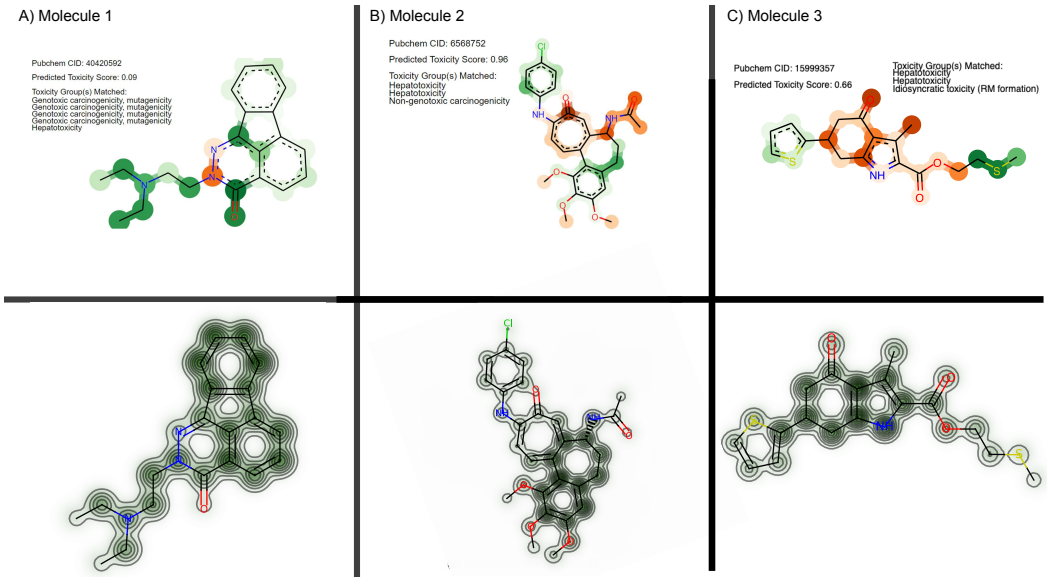


Figure 2.6: Cytotoxicity case studies. Three molecules from the FMP dataset are visualized, using either their attention scores from ToxSmi (top) or their cytotoxicity maps using deep Taylor decomposition (bottom) following the original work [98]. For ToxSmi, the color mapping is identical to Figure 2.4: green for toxicophore atoms and orange for non-toxicophore atoms. Opacity corresponds to the attention score.

dicted as cytotoxic ($\hat{y} = 0.96$) and ToxSmi partly relied on a hepatotoxicity alert from 4-Ethylphenol that was also identified by *Webel* et al. [98]. The third molecule (Figure 2.6**c**) is especially interesting. While it was correctly predicted as toxic by ToxSmi, both models did not highlight a particularly challenging substructure, namely the Thiophene ring, which is sometimes associated to idiosyncratic drug reactions [40, 133]. However, ToxSmi largely based its correct prediction on the Sulfur tail, a hepatotoxicity toxicophore [40] that was not identified by *Webel* et al. [98].

Overall, it has to be emphasized that this analysis relied purely on *unsupervised* learning of toxicophores; neither our nor the model by *Webel* et al. [98] is aware of the notion of toxicophores. While their work mostly focused on the potential identification of new toxicophores, we validated our method in light of existing toxicophores. However, the

dark red shaded areas in our attention maps might give a good starting point in the search for new toxicophores.

2.11 Discussion

In this section, we have conducted an extensive comparison of different molecular representations and machine learning models for toxicity prediction. The experiments revealed that competitive performance can be achieved with purely sequence based chemical language models that do not rely on traditional descriptors (such as fingerprints) or structural or topological features. Moreover, we find that SELFIES [65], a chemical language devised for generative modeling, exhibits comparable predictive power for QSAR tasks to SMILES and similarly benefits from augmentation.

Importantly, we presented *ToxSmi*, a simple and interpretable model that relies solely on SMILES sequences. Coupling ToxSmi with SMILES augmentation, we surpassed a wide range of previous models and obtained SOTA performance on several QSAR tasks, including but not limited to toxicity. Compared to graph-based models [36, 105] ToxSmi is a simplistic model that solely relies on chemical languages such as SMILES and exploits data augmentation to boost model performance and outperforms almost all previous work. Compared to GROVER [36], a larger model that consistently outperformed ToxSmi, an advantage of ToxSmi is that it does not require large-scale pretraining and is thus particularly suitable for low data/resource settings. A key feature of the proposed model is the self-attention mechanism, an ante-hoc interpretability method that learns to extract the most important chemical motifs without explicit supervision. On the Tox21 dataset, we demonstrated that the attention on toxicophores is significantly enriched compared to remaining chemical motifs. These attention maps can not only be useful to validate existing toxicophores but also support in the potential identification of unknown toxicophores. We also evaluated two simple methods for uncertainty estimation that can not only help identifying misclassified samples but only form an implicit model ensemble that further boosts performance. Last, we validated ToxSmi on a proprietary toxicity dataset from *Lisurek* et al. [97] where we found that ToxSmi consistently outperformed a previous model [98] while enabling similar interpretability analyses.

3 Proteochemometrics

3.1 Protein-ligand binding affinity prediction

Proteins are the fundamental building blocks of cellular metabolism and involved in all forms of life. They are vital for our advances in medicinal chemistry since the vast majority of FDA-approved compounds (over 80%) reach their effect by targeting specific proteins [134]. Hence it is evident, that better models of compound-protein interaction (CPI) or protein-ligand binding are instrumental to accelerated molecular discovery.

Background. In the past years, the availability of high-throughput screening (HTS) data for CPI [135] has led to a myriad of novel models for protein-ligand binding (for a review see *Parks* et al. [136]). Traditionally, models were developed on a per-target (or per-assay) basis [137, 138]. Instead, multi-target models employ multi-label classification to combine multiple targets into one model. This approach benefits from cross-target learning [39, 139] which is particularly important in low-data regimes [140] or if tasks (i.e., targets) are correlated [141]. However, since these models do not consider protein descriptors they can only predict binding for new ligands, but not for unseen targets [142, 143].

Instead, proteocheomometrics is concerned with developing models combining features from proteins and ligands [144]. These bimodal models, preferably using deep learning techniques, have become the de-facto standard in protein-ligand binding affinity models [145] and can now be trained large-scale [146, 147] These models bear the advantage that they can, in principle, perform predictions for the entire protein-ligand space. They can learn, for example, inter-molecular non-covalent interactions [121]. We emphasize, however, that generalizing to pairs where both target and ligand is unseen is extremely challenging.

In the first proteochemometric deep learning model, published in 2016, *Tian* et al. [148] took interaction pairs from the STITCH database [149] and used ligand and protein fingerprints to train a simple feedforward network. Today, two major realms can be identified: sequence-based approaches relying on SMILES and protein primary structure [52, 119, 121, 150] and structure-based approaches that either apply 3D CNNs to the binding site [151, 152, 153] or GNNs on molecules and protein secondary or tertiary structure [120, 154, 155, 156, 157]. While one might conjecture that structure-based methods model binding dynamics more realistically, their practical superiority still has to be

demonstrated thoroughly for this task. Very recently, *Volkov* et al. [158] reported that incorporating non-covalent interactions does not improve binding affinity prediction performance compared to simple protein/ligand descriptors

Notably, in the recently conducted IDG-DREAM challenge about Drug-Kinase Binding Prediction [146], the winning method (out of $\sim$ 100 submissions) was a multimodal language model relying on SMILES and amino acid sequences.

3.1.1 Scope of this chapter

In this chapter we propose two novel affinity prediction models. First, a proteochemometric, sequence-based language model called BiMCA which is a bimodal extension of the previously introduced ToxSmi model. Secondly, we propose a simple, yet novel bimodal k-NN model for the same task. Initially conceived as a baseline model we report competitive performance of this model. We first study these models on general protein-ligand binding affinity prediction and compare them to various prior art. We then perform an in-depth investigation on protein kinases, arguably the most important protein family in drug discovery. We challenge the common belief that full protein sequence information is necessary to develop strong affinity predictors and demonstrate that superior performance can be achieved using a tiny subset of the residues only. Last, we introduce novel sequence augmentation mechanisms that can be generally applied in protein language models that rely on a subset of residues from the full proteins.
Note that we develop these models so that they can later be used to evaluate the performance of molecular generative models conditioned on protein sequences.

3.2 Problem formulation

Let $\mathcal{P}$ denote the space of proteins, $\mathcal{M}$ the molecular space and $\mathcal{A}$ the affinity scores. We are interested in learning a function $\Phi_A : \mathcal{P} \times \mathcal{M} \rightarrow \mathcal{A}$. The function Φ_A maps a protein-ligand tuple to an affinity score and is learned from the training data set $\mathcal{D} = \{p_i, m_i, a_i\}_{i=1}^{N}$ where $p_i \in \mathcal{P}, m_i \in \mathcal{M}$ and $a_i \in \mathcal{A}$ is the scalar binding strength, denoted by the pIC50; the negative log of the the half-maximal inhibitory concentration (IC50).

3.3 Proposed models

3.3.1 BiMCA – A proteochemometric language model

To address the presented multimodal regression problem, we propose the BiMCA, a Bimodal Multiscale Convolutional Attention model. The BiMCA is similar to the one we conceived earlier in *Manica*, *Oskooei*, *Born*, et al. [27] for drug-sensitivity prediction. For an

overview see Figure 3.1. This model ingests a molecule $m \in \mathcal{M}$ represented as SMILES

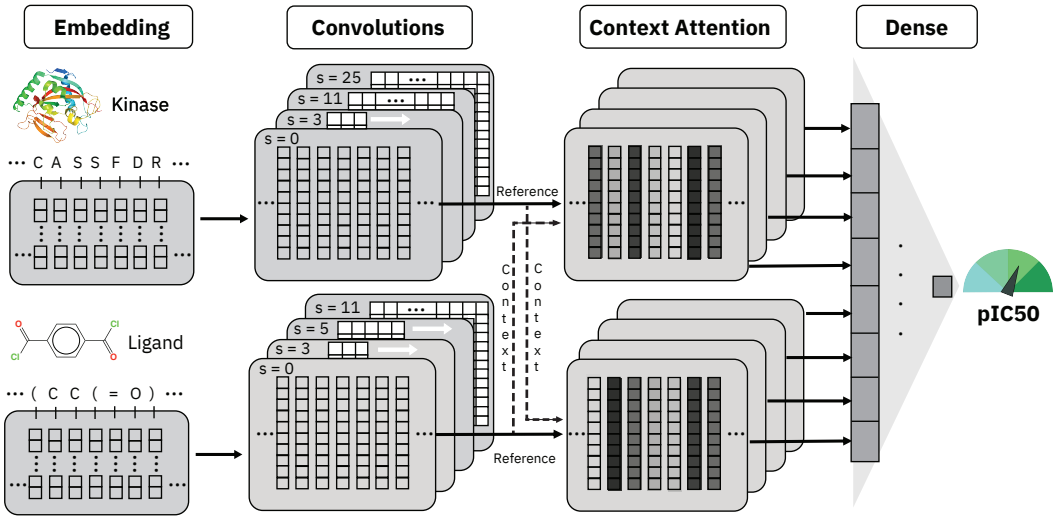


Figure 3.1: The bimodal multiscale convolutional attention model (BiMCA). Proteins and ligands are represented as text sequences of amino acids and SMILES respectively. The BiMCA uses learned embeddings and then applies 1D-convolutions with varying kernel sizes on the embedding matrices. Afterwards, the context attention layers fuse information from both modalities and generate the attention scores over one input modality, using the other modality as context. Black arrows show the information flow through the network, white arrows the direction of the convolution sliding.

string and a protein $p \in \mathcal{P}$ represented by its primary structure. SMILES sequences are tokenized and padded to a length of $T_M = 696$ and each token is represented as a learned embedding of dimensionality $H_M = 32$, s.t. the input matrix $\boldsymbol{X}_M \in \mathbb{R}^{T_M \times H_M}$. Proteins are left-padded to a length of $T_P = 2536$ and each token is represented as a learned embedding of dimensionality $H_P = 8$, s.t. the input matrix $\boldsymbol{X}_P \in \mathbb{R}^{T_P \times H_P}$. Three parallel channels with convolutions of kernel sizes 3, 5 and 11 and 3, 11 and 25 are employed on the SMILES and protein sequences, respectively. In both cases, a fourth channel has a skip connection without convolutions.

For each of the four channels in both modalities, we have one attention layer, accounting for a total of 8 layers. In these layers, broadly speaking, one modality is used as a context to compute the attention scores for the reference modality. The output of the largest convolution kernels of the kinase stream (size 25) is coupled with the output of the largest kernels of the ligand stream (size 11) and so on. This mechanism is the bimodal extension of the self-attention mechanism in ToxSmi (cf. Equation 2.1). It allows the model to use

information from the binding partner (context) in learning the importance of each token in the input sequence (reference). The attention weights α_i are computed as:

$$\alpha_i = \frac{\exp(u_i)}{\sum_j^T \exp(u_j)} \quad \text{, where} \quad \vec{\mathbf{u}} = \tanh(\mathbf{X}_1\mathbf{W}_1 + \mathbf{W}_3(\mathbf{X}_2\mathbf{W}_2))\vec{\mathbf{v}} \tag{3.1}$$

We call $\mathbf{X}_1 \in \mathbb{R}^{T_1 \times C}$ the *reference* input, where $T_1 \in \{T_M, T_P\}$ is the sequence length and C is the number of convolutional filters. Further, $\mathbf{X}_2 \in \mathbb{R}^{T_2 \times C}$ is the *context* input, where $T_2 \in \{T_M, T_P\}, T_1 \neq T_2$ is the sequence length in the other modality. $\mathbf{W}_1 \in \mathbb{R}^{C \times A}$, $\mathbf{W}_2 \in \mathbb{R}^{C \times A}$, $\mathbf{W}_3 \in \mathbb{R}^{T_1 \times T_2}$ and $\vec{\mathbf{v}} \in \mathbb{R}^A$ are learnable parameters. It is identical to Bahdanau attention apart from $\mathbf{W}_3$ which we need since $T_M \neq T_P$. Intuitively, both inputs are projected into a common attention space $\mathbb{R}^A$ and then summed up, which enables the layer to take the context into account for determining feature relevance. $\vec{\mathbf{v}}$ combines the information through a dot product, the output of which is fed to a softmax layer to obtain the attention weights α_i, which are used to filter the inputs, like in ToxSmi's self-attention (cf. Equation 2.2). We call this mechanism **context-attention**, a visualization can be found in Figure 3.2. The filtered protein/ligand information gets

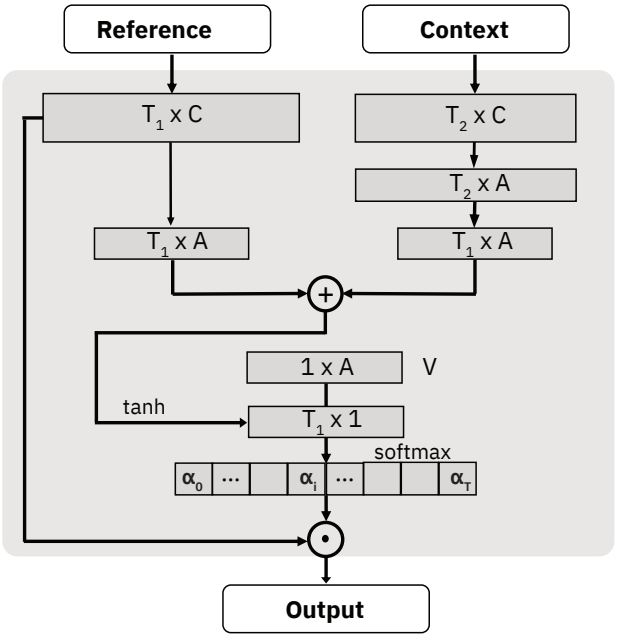


Figure 3.2: The context attention layer in the BiMCA model. This layer receives two inputs, the *context* and the *reference*. Both inputs are projected into a joint attention space $\mathbb{R}^A$ and summed up in that space. The final softmax produces the attention weights $\alpha_i \in [0, 1] : \sum_i \alpha_i = 1$.

passed to a stack of dense layers which outputs the predicted pIC50.

Hyperparameters and training procedure

The hyperparameters can be found in Table 3.1. All models were implemented in `PyTorch` [159]. We optimized a MSE loss with Adam [77] and trained for 50 epochs with a learning rate of 5e−3, a batch size of 128 on a cluster equipped with `POWER8` processors and a single `NVIDIA Tesla P100`.

Parameter	Value
Protein sequence length T_P	2536
SMILES sequence length T_M	696
Protein embedding size H_P	8
SMILES embedding size H_M	32
Protein 1D conv. kernel sizes	$[3, 11, 25]$
SMILES 1D conv. kernel sizes	$[3, 5, 11]$
Number of protein kernels	$[32, 32, 32]$
Number of SMILES kernels	$[32, 32, 32]$
Protein attention size A	16
SMILES attention size A	16
Dense layer sizes	$[64]$
Activation function	ReLU
Dropout	30%
Learning rate	3e−4

Table 3.1: Hyperparameters of the BiMCA.

3.3.2 k-Nearest-Neighbor

We seek to compare the BiMCA to an alternative model relying on more traditional machine learning methods. Due to its simplicity and ease of interpretation, we propose a k-NN that computes distances in a joint space spanned by proteins and ligands. To the best of our knowledge, the only previous report of a *bimodal* k-NN for affinity prediction is from *Nazarshodeh* et al. [160] who used numerical descriptors. In contrast, here we represent kinases by their primary structure (either full sequence or only active site) and molecules by their ECFP4 fingerprint [46] with 512 bits. As a distance metric between samples we utilize a combination of the length-normalized Levenshtein distance [161] for the primary structure and the Tanimoto similarity [162] of molecules. More formally, let $\{p_j, m_j, a_j\}$ denote an unseen sample from the test dataset $\mathcal{D}_{Test}$. With the goal of predicting $\hat{a}_j$ to approximate the unknown a_j, we first retrieve the subset of training data $\mathcal{D}_k$ containing the k nearest neighbors using the distance measure

$$\mathbf{D}(p_i, m_i, p_j, m_j) = \frac{Lev(p_i, p_j)}{max(|p_i|, |p_j|)} + (1 - \mathcal{T}(m_i, m_j)) \tag{3.2}$$

where $|\cdot|$ denotes sequence length, $\mathcal{T}$ is the Tanimoto similarity [162] and $Lev(\cdot,\cdot)$ is the Levenshtein distance [161]. The Levenshtein distance is a string-based distance measure that, in this context, counts the number of single-residue transformations (insertions, deletions or substitutions) required to transform one sequence into the other. Note that:

$$Lev(p_i, p_j) \in [0, \max(|p_i|, |p_j|)] \tag{3.3}$$

and

$$\mathcal{T}(m_i, m_j) \in [0, 1] \tag{3.4}$$

hence both addends of Equation 3.2 are scaled to the same range, i.e., they have equal importance[1]. Hence, $\mathbf{D}(\cdot,\cdot,\cdot,\cdot) \in [0, 2]$. Then, the prediction $\hat{a}_j$ is trivially computed by $\hat{a}_j = \frac{\sum_i^k a_i}{k}$ with $a_i \in \mathcal{D}_k$. As k-NN is a lazy learning method, the inference runtime scales with the dataset size and one query can thus easily require to compute hundred thousand distances. Therefore, in practice we compute $\mathbf{D}$ not for all training samples but only for those samples $\{p_i, m_i\}$ where either 1) $p_i = p_j$, 2) $m_i = m_j$ or 3) p_i is one of the 10 most similar sequences to p_j in the training dataset.

3.4 Datasets and preprocessing

3.4.1 Data splitting strategies

The bimodal nature of the problem formulation of proteochemometric models gives rise to four complementary paradigms in evaluating model generalization. These paradigms are essentially determined by the splitting procedure that is used to separate training and testing data (cf. Figure 3.3). In a naïve splitting strategy, paired samples consisting of two modalities (here: a protein and a drug) are build and thereafter randomly split into train and test samples. This is not only the most lenient, but also the most commonly adopted strategy and it imposes a high risk that the model can perform ostensibly well by merely memorizing training samples (as only the *pair* is unseen, whereas both the molecule and the protein might be known). We call this the **lenient split**.

Instead, in the classical drug discovery setting, it is desirable to generalize to new molecules (cf. Figure 3.3 *top right*). We call this the strict drug split or just **drug split** or **ligand split**. A more challenging variant of this split is the **scaffold split**, where ligands are grouped together by their scaffold (usually the Murcko scaffold). The scaffolds are randomly split between train and test, thus ligands with the same scaffold always end in the same data partition, ensuring a higher dissimilarity between train and test ligands.

Instead, in a drug repurposing setting the objective is to find a target which is strongly inhibited by a compound. Hence, it is desirable to generalize to new proteins. We call this the strict protein split or just **protein split**. A more challenging variant of this split

[1] We refrain from extending this with a simple weighting scheme for the two distances.

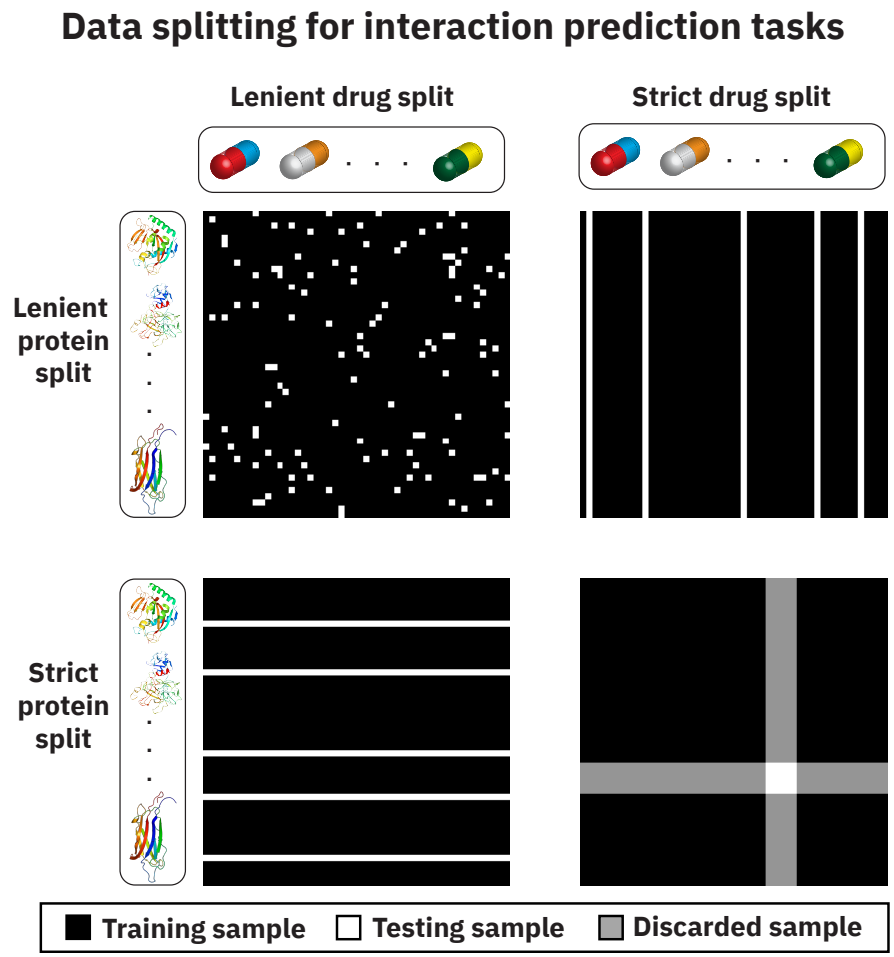


Figure 3.3: Data splitting strategies for binding affinity prediction. The four possible strategies to split samples between training and testing data are shown.

is the **protein family split** where, similar to the scaffold split, the proteins are not split directly, but rather the protein families are split before all samples associated to a protein of a family are assigned to the respective data partition.

While these regimes are certainly more challenging, a *strict* splitting strategy (i.e., stratifying by both modalities simultaneously, cf Figure 3.3 *lower right*) represents the most challenging task. Such a split is better suited to assess whether a model actually learned generic features of protein-ligand binding. The long-term objective is to build interaction prediction models that can extrapolate to the entire protein-ligand space – meaning they can accurately predict binding for unseen drugs screened against unseen protein. We call this the **strict split**.

Splitting the data meaningfully is especially relevant in light of the hidden ligand bias [163], i.e., the observation that binding affinity predictions are mostly based on ligand rather than interaction features [150].

3.4.2 DeepAffinity dataset

BindingDB [164] is the largest publicly available resource of protein-ligand binding data. To facilitate direct comparison with previous work, we did not retrieve the raw data from BindingDB but rather relied on the data processed and split by the authors of DeepAffinity [119]. This dataset comes with $263,583$ training and $113,168$ test samples in a *lenient split*. Additionally it contains four held-out sets to test the generalization to entirely unseen protein families, namely: Nuclear estrogen receptors (ER; $34,318$ samples), ion channels ($14,599$ samples), receptor tyrosine kinases (RTK; $34,318$ samples) and G-protein coupled receptors (GPCR; $60,238$ samples).

Models are evaluated with root-mean-squared error (RMSE) and Pearson correlation coefficient (PCC) between the predicted and true pIC50 which is roughly in the range 3 to 10 (1mM to 0.1nM).

3.5 Quantitative comparison to prior art

Our first objective is to compare the performance of the BiMCA and k-NN model with previous proteochemometric approaches for protein-ligand affinity prediction.

3.5.1 Lenient split

The results of our BiMCA and k-NN models on the lenient BindingDB split as provided by *Karimi* et al. [119] are shown in Table 3.2.

Model	RMSE	PCC
DeepDTA [52]	0.782	0.848
DeepAffinity [119]	0.780	0.840
DeepCDA [165]	0.808	0.844
MONN [121]	0.764	0.858
NN (k=1)	0.862	0.83
k-NN (k=4)	**0.728**	**0.871**
k-NN (k=13)	0.783	0.848
BiMCA (full seq.)	0.892	0.786

Table 3.2: Performance comparison for different models on fixed-split BindingDB dataset from *Karimi* et al. [119]. All models were trained and evaluated on the same samples. Models below the dashed line are ours.

Surprisingly, the simplistic k-NN model achieved the best results on this split and outperformed all SOTA methods such as MONN [121], DeepAffinity [119] or DeepDTA [52]. This emphasizes the necessity to compare any new, complex architecture to a baseline, especially in bimodal tasks like affinity prediction where one modality alone might carry most of the predictive power (e.g., the hidden ligand bias [163]). However, it also questions the usability of a lenient splitting strategy. The k-NN solely predicts based on distances to training samples, computed separately in a protein and a ligand space. Given that it perfectly memorizes the training data, it is less surprising to see it performing well in a lenient split.

Moreover, we emphasize that the authors of DeepDTA, DeepAffinity and MONN all build model ensembles consisting of up to 30 individual models. These models achieved better results than the individual models shown in Table 3.2 (up to RMSE of 0.658 and PCC of 0.895 [121]), but to ensure a fair comparison with our single models (it is widely known that model ensembles improve performance [166, 167]), we omitted all ensembles from the results in Table 3.2.

Notably, the performance of the BiMCA model on the lenient split falls behind the listed SOTA models on the lenient split (cf. Table 3.2). This is unsatisfying but will be examined further in the next experiment.

3.5.2 Protein family split

In this split, we used the same models to test the generalization ability to unseen protein families, namely Ion Channels, GPCRs, RTKs and Estrogen Receptors. The results can be found in Table 3.3 and show that the BiMCA model excels at the generalization to novel protein families and outperforms all previous models we found in the literature on three out of four protein families. Interestingly, only in the generalization task to Receptor Tyrosine Kinases (RTK) the BiMCA is outperformed by previous work. This is remarakble given that we also included the ensemble models from *Karim* et al. [44] and *Truong Jr* [168] in the comparison.

Model	ER		Ion Channel		RTK		GPCR		Mean	
	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC
DeepAffinity SMILES [119]	1.53	0.16	1.34	0.17	1.24	0.39	1.40	0.24	1.38	0.24
DeepAffinity Graph [119]	1.68	0.05	1.43	0.10	1.74	0.01	1.63	0.04	1.62	0.05
DeepCDA [165]	-	0.10	-	0.31	-	0.42	-	0.28	-	0.28
ECFP/Pfam-based [168]	1.74	0.19	1.32	0.27	1.27	0.43	1.49	0.22	1.46	0.28
DeepAffinity Ensemble [119]	1.46	0.30	1.30	0.18	1.23	0.42	1.36	0.30	1.34	0.30
MLP ensemble [168]	1.51	0.24	1.36	0.19	1.26	0.42	1.36	0.33	1.37	0.29
Transformer ensemble [168]	1.61	0.39	1.34	0.38	**1.14**	**0.47**	1.29	0.33	1.35	**0.39**
NN (k=1)	1.53	0.30	1.80	0.07	1.51	0.32	1.81	0.17	1.66	0.22
k-NN (k=4)	1.36	0.30	1.52	0.11	1.31	0.37	1.50	0.20	1.42	0.25
k-NN (k=13)	1.28	0.40	1.43	0.13	1.26	0.36	1.43	0.17	1.35	0.27
k-NN (k=25)	**1.27**	**0.43**	1.41	0.13	1.25	0.34	1.42	0.15	1.33	0.26
BiMCA (full seq.)	1.35	0.32	**1.19**	**0.41**	1.38	0.40	**1.25**	**0.42**	**1.27**	**0.39**

Table 3.3: Generalization to new protein families based on fixed-split BindingDB dataset from *Karimi* et al. [119]. DeepAffinity models refer to unified RNN-CNN and RNN/GCNN-CNN models. All models below the single line are ours. Models below the dashed line and above the regular line are ensembles which can hardly be directly compared to our models. Numbers from other works taken from their manuscripts since the split is fixed. DeepCDA did not report RMSE. The last two columns report the mean across the four datasets.

3.6 Human kinases - finding compact protein representations

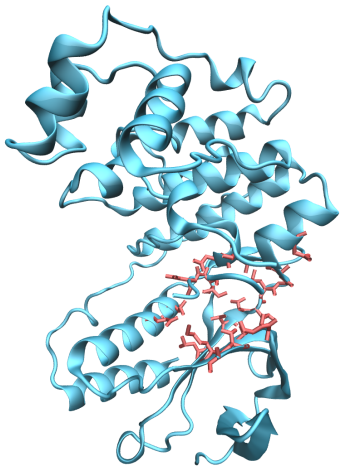
3.6.1 Motivation

In the previous section, we have seen that the BiMCA, a proteochemometric language model can reach SOTA performance in binding affinity prediction for unseen protein families. One critical aspect in our own as well as all previously proposed sequence-based models (e.g. [44, 52, 120]) is that they rely on the full protein primary structure, i.e., the entire amino acid sequence of a protein.

From a computational perspective this is disadvantageous since large proteins or protein complexes can easily consist of > 2000 residues. In practice, this implies that many sequence-based models use a tremendous amount of zero-padding or simply remove large proteins from the training corpus.

Much more importantly, using the entire primary structure deeply conflicts with our knowledge about protein-ligand binding. It is well understood that binding behaviour is not governed by the entire protein but rather by their binding pockets [169]; the first account for that dates back to 1894 [170]. Therefore, in theory, it should be sufficient to provide a subset of residues for accurately predicting binding. But in practice, for a general protein it is largely unclear which residues correspond to potential binding pockets, especially if protein tertiary structure has not been determined experimentally.

In this section, we will attempt to find a more compact protein sequence representation for binding affinity prediction. We hypothesize that we can replace the full sequence with a much shorter sequence (comprising only a tiny set of residues that govern the binding behavior) without reducing predictive accuracy. This hypothesis will be assessed on a specific protein family, protein kinases, which constitute a key protein family for cell metabolism. As an alternative representation to the full protein, we will rely on the kinase ATP-binding site, the binding pocket addressed by most kinase inhibitors [171]. In the following, we will refer to this as *active site*. Since the active site sequence is composed of discontiguous subsequences of the full primary structure (cf. Figure 3.4), the use of such a condensed representation offers an implicit way of incorporating 3D information into a 1D model.



Human Kinase

Full Sequence

ITMKHKLGGGQYGEVYEGVWKKYSLTVAVKTLKEDTMEVEE
FLKEAAVMKEIKHPNLVQLLGVCTREPPFYIITEFMTYHNLLD
YLRECNRQEVNAVVLLYMATQISSAMEYLEKKNFIHRDLAAR
NCLVGENHLVKVADFGLSRLMTGDTYTAHAGAKFPIKWTAP
ESLAYNKFSIKSDVWAFGVLLWEIATYGMSPYPGIDLSQVYE
LLEKDYRMERPEGCPEKVYELMRACWQWNPSDRPSFAEIH
QAFET

Active site only

LGGGQYGEVAKEMVITEFMTYGDRNLADL

Figure 3.4: Full kinase versus active site sequence representation. While the active site forms a localized binding pocket in the tertiary structure (left), the residues comprising the active site lie discontiguously in the full primary structure (right).

We believe that, so far, no other work has systematically assessed the impact of using active site rather than full protein sequences in proteochemometric binding affinity prediction models. Here, we aim to fill this gap. The comparison will be performed using the two models defined in Section 3.3, the BiMCA and the k-NN. The BiMCA was selected because we wanted to examine a SOTA language model. Due to the competitive performance of the k-NN in Section 3.5, we decided to also examine the k-NN, especially because it is an ante-hoc interpretable model which, unlike the BiMCA, is not prone to diluting simple trends in the data behind countless non-linearities.

3.6.2 Protein kinases

Since the FDA-approval of imatinib (2001), the first marketed kinase inhibitor, kinases have arguably become the most potent source of targets for drug discovery [172, 173, 174]. By 2021, about 60 kinase inhibitors are marketed and expand treatment options for cancer and, more recently, also neurodegenerative or viral diseases [175]. The characteristics of the target family that led drug discovery researchers to avoid kinases for many years (e.g., binding site similarity and sheer size [172]) make the family an ideal candidate for proteochemometric approaches – since they exploit this similarity systematically. In the past years, computational approaches have advanced our understanding of kinases on e.g., identification of binding subpockets [176] or promiscuity maps [177], inhibitor selectivity [178], defining the kinome conformational space [179] or virtual screening such as drug response [180, 181] or CPI prediction [137, 138, 182, 183].

3.6.3 Data preprocessing and training setup

3.6.3.1 BindingDB dataset

To assess the posed research question, we curated protein-ligand binding data from BindingDB [164], a large but heterogeneous data source that comprises data scraped from publications as well as other databases such PubChem [4] and thus does not follow a standardized experimental procedure.

From the $2,222,074$ entries of the database as on 22.04.2021, $\sim$800,000 were retained after removing missing values and duplicates. Afterwards, samples with molecules whose SMILES strings were invalid or longer than 696 tokens, i.e. atoms and/or bonds, were removed. In alignment with previous work [52, 119, 121] and our results shown above, we chose IC50 as binding affinity metric, convert all values to pIC50. The values were clipped to the interval $[2, 11]$ (1mM to 0.01nM). Last, we filtered out all samples where the target proteins are not human kinases. This resulted in $206,989$ samples distributed across $113,475$ ligands (mean pIC50 per ligand: 7.1 ± 1.2) and 349 human kinases (mean pIC50 per kinase: 6.2 ± 0.9). See Figure 3.5 for an overview of the dataset's statistics. A significant bias in BindingDB is that more deeply studied kinases (i.e., kinases that were screened against more ligands) tend to have a higher average binding affinity ($r = 0.39$).

Non-kinase data. The rest of the above data (i.e., all non-kinome samples) make up $485,461$ samples which are distributed across 2856 proteins and $331,169$ ligands. This data is used in one configuration for pretraining the BiMCA.

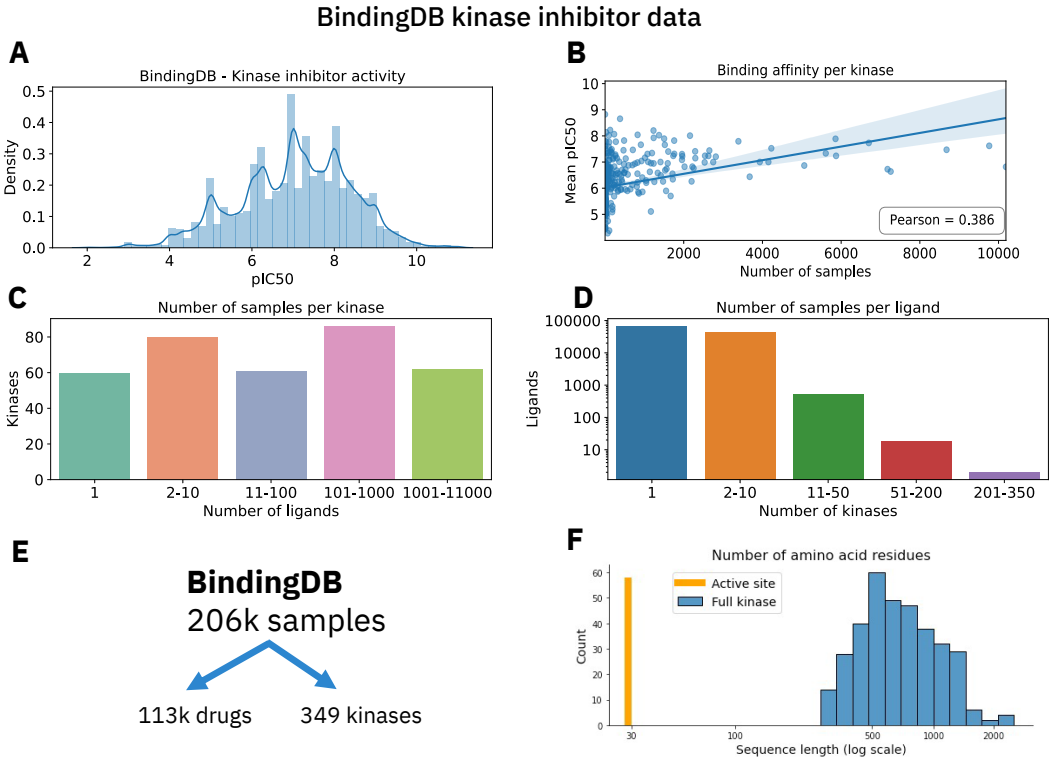


Figure 3.5: Visualization of kinase inhibitor data in BindingDB [164]. **A)** pIC50 distribution for kinase samples ($N = 206,989$). **B)** Kinases with more affinity samples tend to have higher average binding affinity (i.e., are more promiscuous). **C)** Histogram of number of data points for each kinase. **D)** Most ligands are screened on less than a dozen of kinases but some are screened against almost all 349 kinases. **E)** Division of 206k BindingDB samples into unique kinases and inhibitors. **F)** Distribution of sequence length of full sequences and active sites (log-scale).

3.6.3.2 Human kinase sequence alignment

To extract the active sites for the human kinases, we relied on the binding site definition of protein kinase A (PKA) proposed by *Sheridan* et al. [183]. Their work identified 29 residues which constitute the ATP binding site.

Definition 3.6.1 (*Sheridan* active site [183])**.** The 29 residues comprising the Sheridan active site are defined as follows. $\mathcal{R}_{\text{Sheridan}} = \{$ A70, D184, 121, E170, E91, F187, F54, G125, G126, G50, G52, G55, K72, L173, L49, L95, M118, M120, N171, P124, R56, S130, S53, T183, T51, V104, V123, V57, Y122. $\}$

In brief, their alignment consisted of three steps:

1. Taking existing, reliable multiple-sequence alignment (MSA) of kinases within eight human kinase groups [184],

2. Finding correspondences between active site residues of different groups from 3D superpositions of kinases,

3. Pooling this with the MSAs of 1.

These resulting residues include contributions from the Gly-rich-loop, gatekeeper, hinge, and DFG-in-out. We then applied these 29 residues to the structurally-validated MSA of 497 human kinases from *Modi* and *Dunbrack* [185]. The obtained active site sequences are more than an order of magnitude shorter than the full sequences (cf. Figure 3.5**F**).

3.6.3.3 Data splitting

We explored two data splitting strategies (cf. Figure 3.3):

1. **Ligand split**: For this split, we put aside the samples associated to 10% of the ligands. Then, we conducted a 10-fold cross-validation (CV) on the rest. All splits were stratified by the number of samples as well as the mean pIC50 per ligand.

2. **Kinase split**: This split assessed the generalization power toward unseen kinases. Again, we put aside 10% of the kinases and then conducted a 10-fold CV on the remainder of the data. Again, all splits were stratified by the number of samples as well as the mean pIC50 per ligand.

Pretraining. The $485,461$ remaining, non-kinase samples were split with a $90/10$ ratio into train and test data. This data was used in an additional configuration where the BiMCA was pretrained on non-kinase data. Notably, ligands in the pretraining dataset were not excluded from the cross-validation in the ligand split in order to keep the kinase dataset sufficiently large and guarantee perfect comparability between kinase and ligand split. This implies that 3.5% of the ligands (accounting for 5.6% of the samples) in the kinase dataset were already seen during pretraining. Notably however, as kinases were held out from the pretraining data, these ligands were only presented together with non-kinases. We validated that this did not positively impacted performance of the pretrained BiMCA.

3.6.3.4 Hyperparameters and model training

k-NN. This model was defined in Subsection 3.3.2. The k-NN was evaluated on all $k \leq 25$. For all results, we choose a value of $k = 13$ as this led to the lowest RMSE on the validation dataset on the ligand split.

BiMCA. This model was defined in Subsection 3.3.1. The hyperparamters are given in Table 3.4. The differences in the number of convolution kernels and attention sizes are done to partly compensate for the fact that the full sequence model had substantially more parameters than the active site model. This stemmed from the context attention layer which requires $\mathcal{O}(nm)$ parameters where n and m are the sequence length of proteins and ligands respectively. In total, the active site model only consisted of 651k parameters, less than 5% of the full sequence model (14M). All models were implemented in `PyTorch` [159]. The BiMCA model optimized a MSE loss with Adam [77] and was trained for 50 epochs with a batch size of 128 on a cluster equipped with `POWER8` processors and a single `NVIDIA Tesla P100`.

Parameter	Full sequence	Active site
Protein sequence length T_P	2536	32
SMILES sequence length T_M	696	696
Protein embedding size H_P	8	8
SMILES embedding size H_M	32	32
Protein 1D conv. kernel sizes	$[3, 11, 25]$	$[3, 11, 25]$
SMILES 1D conv. kernel sizes	$[3, 5, 11]$	$[3, 11, 25]$
Number of protein kernels	$[32, 32, 32]$	$[128, 128, 128]$
Number of SMILES kernels	$[32, 32, 32]$	$[128, 128, 128]$
Protein attention size A	16	64
SMILES attention size A	16	64
Dense layer sizes	$[64]$	$[200]$
Activation function	ReLU	ReLU
Dropout	30%	30%
Learning rate	5e−4	5e−4

Table 3.4: Hyperparameters of the BiMCA for both models.

3.6.4 Learning binding from full proteins vs. active sites

3.6.4.1 Kinase data split

Aggregated performance results. This split is best suited to test the influence of the two protein representation types. Since the shape of the ATP-binding pocket largely determines the binding behavior [186], this split is significantly more challenging than a ligand split. As mentioned above, the so-called *hidden ligand bias* [163] refers to the observation that binding affinity predictions are mostly based on ligand rather than protein features or interaction features [150]. Therefore, a kinase split can test the generalization abilities of our affinity models better than a ligand split. As can be seen in Figure 3.6, the results of the 10-fold CV show a consistent and strong superiority of the active site configuration for all three model types (k-NN, BiMCA, BiMCA pretrained). On the validation data, the RMSE is reduced by 1.2%, 7.5% and 6.9% for the k-NN, the BiMCA and the pretrained BiMCA respectively. This is remarkable because the full se-

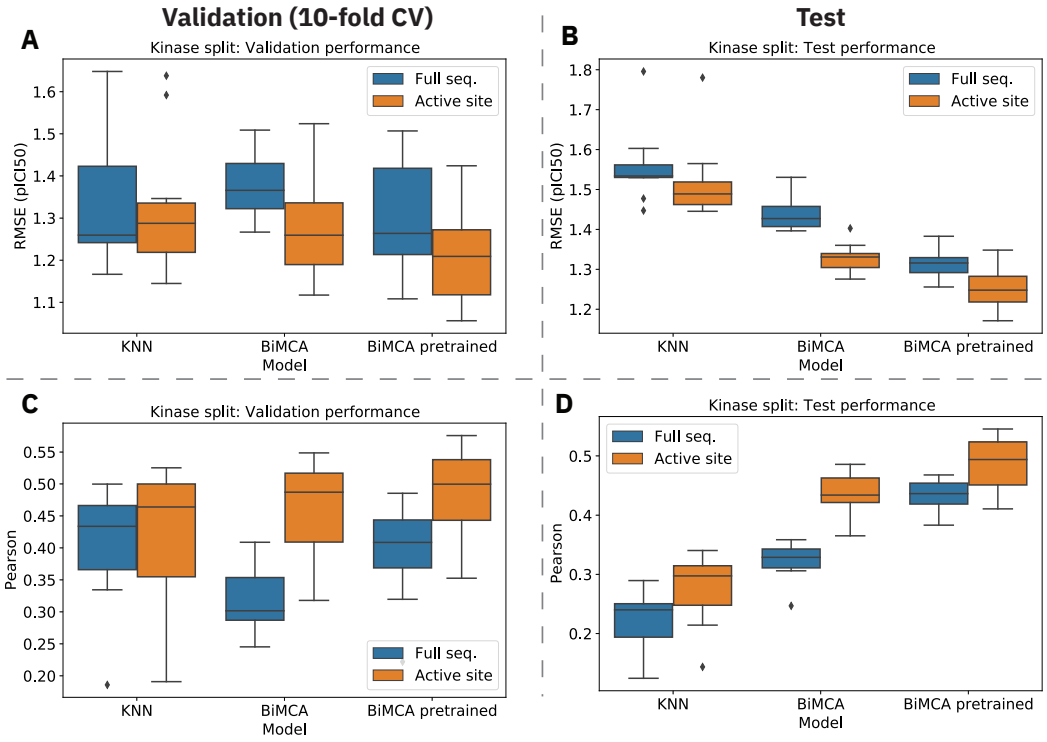


Figure 3.6: Binding affinity prediction results on kinase split. The left and right column show respectively the performance of all three models on the validation and test data. In terms of RMSE as well as PCC, the active site configurations significantly outperform the full sequence configuration. This occurs irrespective of the utilized model. The numerical results can be found in Table A3.1 and Table A3.2.

quence contains an order of magnitude more features (mean sequence length: 742 vs. 29 residues). Moreover, the active site BiMCA models only have $\sim$ 5% of the parameters of the full sequence model. The validation results are less consistent than the test results because of the heterogeneity across the ten folds. Again, high standard error in the performance are expected in this setting [186]. The test results in Figure 3.6**B** and **D**) show that the full sequence configurations achieve an average improvement in RMSE of 2.6%, 7.6% and 4.6%. In all three cases, the full sequence models are outperformed by the active site models ($p < 0.01$, Wilcoxon signed-rank test, $W+$). We also observe that the BiMCA models outperforms the k-NN by a large margin. For the *pretrained* BiMCA setting we used all non-kinase data from BindingDB to warm up the model before fine-tuning on the kinase data. After 20 epochs of pretraining, this model achieved a RMSE of 0.86 and a PCC of 0.82 on the non-kinase data (lenient split). Notably, the pretraining significantly improved performance, demonstrating that learning general patterns of protein-ligand interactions can massively boost the performance of proteochemometric models for kinase affinity prediction. Interestingly, the active site even outperformed the full sequence model although both models were pretrained on full protein sequences.

Grouping kinases. To understand better the performance for the kinome landscape, we assembled eight different groups of conventional protein kinases (ePK) based on the classification by *Hanks* and *Hunter* [184]. With the catalogue from *Manning* et al. [187] that contains $\sim$ 600 kinases, all kinases were mapped to their respective group. For all kinase groups, the PCC is shown in Figure 3.7 for the k-NN and

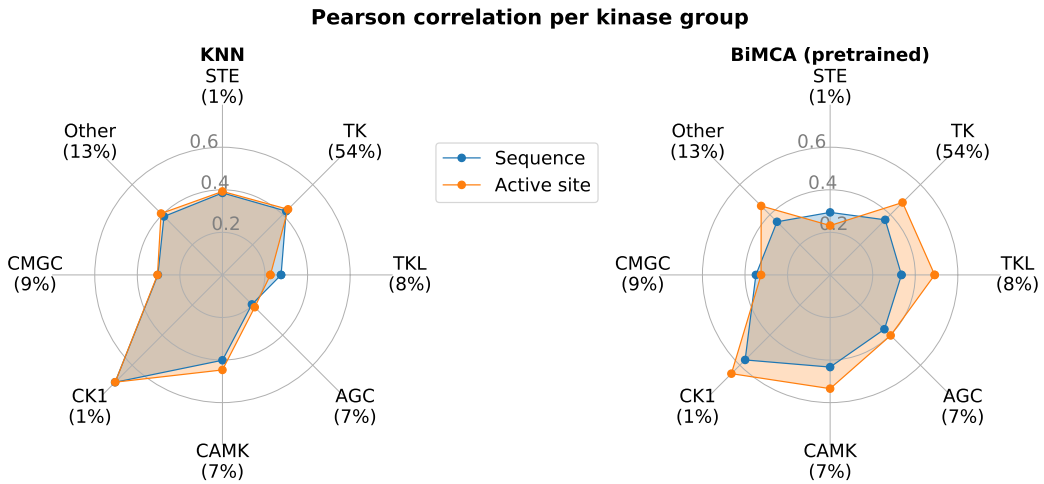


Figure 3.7: Performance in binding affinity prediction for unseen kinases grouped by kinase family. For the k-NN (left) and the pretrained BiMCA (right) the Pearson correlation of all samples of the respective kinase group is shown. Kinases that could not be classified with the catalogue from *Manning* et al. [187] are grouped into *Other*.

pretrained BiMCA model. The plot largely confirms the superiority of the active site configuration consistently across the kinase groups. Most screened kinase-ligand pairs belong to Tyrosine kinases (TK), accounting for 54% of all kinase-related samples in BindingDB. TKs phosphorylate tyrosine residues and are thoroughly researched due to their significant role in cancer and the successful development of highly selective TK inhibitors such as imatinib, gefitinib or erlotinib.

The only exceptions for the superiority of the active sites are the TKL group in the k-NN and the STE and CMGC group in the BiMCA. Let us have a closer look at the TKL (Tyrosine Kinase-Like) group and try to explain why the results do not resonate with the remaining findings indicating superiority of the active site.

The first observation is that for models based on sequence similarity (like the k-NN), full protein structure is superior to active site alone. This could be explained by the fact that many TKL kinases (e.g., all RAF kinases [188]) have multiple binding sites. These are not reflected in the active site sequence which only captures the ATP-binding site. Secondly, the k-NN performs poorly on TKL – there is no other group where the performance gain for the BiMCA compared to the k-NN is higher.

We suspect that this is because TKLs constitute the most heterogeneous group of kinases [187]. Note that the k-NN predicts solely based on sequence similarity. Instead, the BiMCA can capture non-linear relations which might explain the high the performance gain for the BiMCA. Moreover, only the BiMCA can leverage information from distant samples that have more complex relations to the kinase of interest and thus the active site BiMCA configuration achieves the best performance in predicting affinity for unseen TKL kinases.

Remarkable is also the good result for kinases from the CK1 (cell kinase 1) group. We hypothesize that this might be due to the high intra-group and low-inter group similarity of CK1 kinases: CK1s are highly conserved sequences, very similar to each other but very distinct from other kinase groups [189]. In the kinome tree proclaimed by *Manning* et al. [187], CK1s form a distinct branch.

Similarity analysis. A reasonable concern in the kinase split is that the model performance hinges on the availability of similar kinases in the training data. Therefore, we are showing in Figure 3.8 the per-kinase performance as a function of the similarity to the nearest neighbor in the training data. Overall, the plots suggest that our models do not require data from similar kinases to reach their performance. While all PCCs are positive, none of them exceed values of 0.11. Also on this experiment, we observe that the active site configuration gives better results. Note that the best model (BiMCA, active site) has the lowest correlation of all models. Unsurprisingly, the k-NN has a stronger dependence on similar samples than the BiMCA.

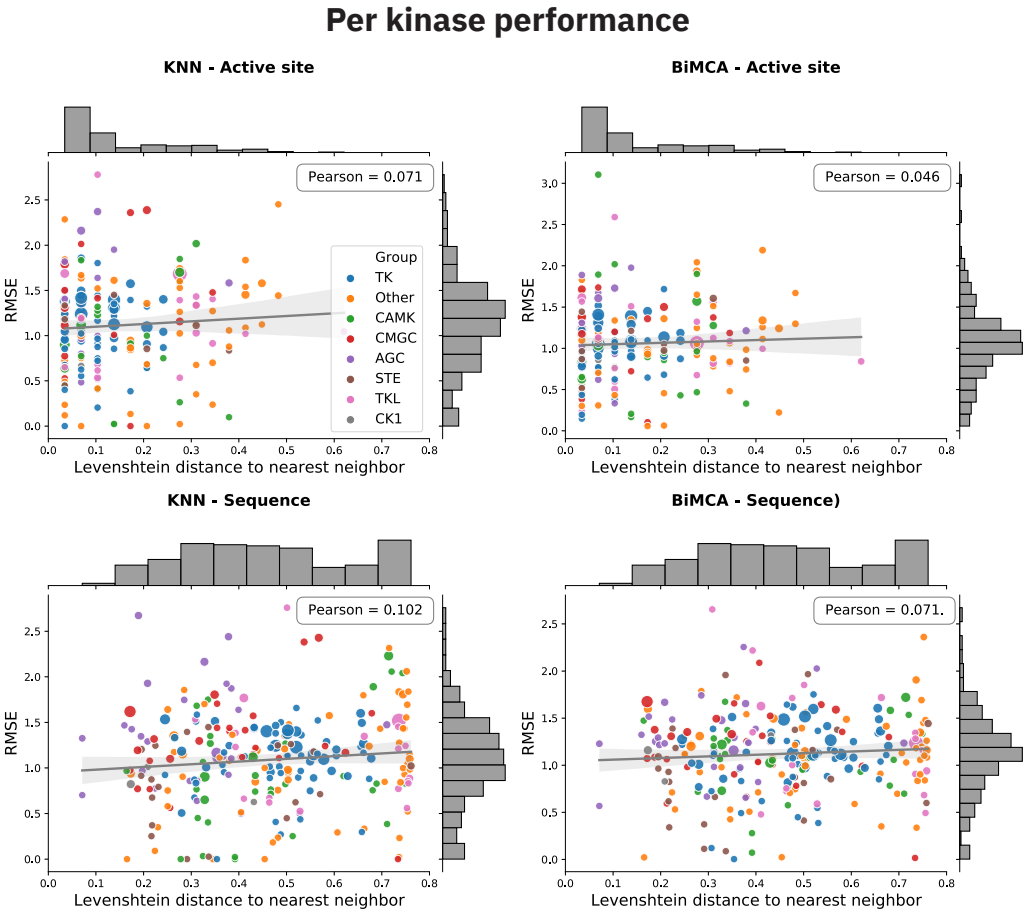


Figure 3.8: Does the predictive accuracy depend on the availability of similar kinases in the training data? No strong correlation between the performance on a specific kinase and the distance to the nearest neighbor in training data was found in any of the four configurations. Plots obtained from validation results.

3.6.4.2 Ligand data split

Aggregated performance results. This split corresponds to the classical drug discovery setting – based on some affinity data for a kinase of interest, the model should predict the potential of a molecule to inhibit this kinase. This task is easier than the kinase-split but still more challenging than a lenient split. The results of the validation and the test data are shown in Table 3.5 (RMSE) and Table 3.6 (Pearson correlation). Like in the kinase split, all BiMCA models using active site information are superior to the ones using full primary structure (8.2% and 4.7% RMSE improvement for the BiMCA and pretrained BiMCA respectively). For both models, these differences are statistically sig-

Data	Configuration	k-NN	BiMCA	BiMCA (pretrained)
Val.	Full seq.	0.78 ± 0.01	0.91 ± 0.01	0.85 ± 0.01
	Active site	**0.77** ± 0.01	**0.83** ± 0.01	**0.82** ± 0.01
Test	Full seq.	**0.76** ± 0.00	0.91 ± 0.01	0.86 ± 0.01
	Active site	0.77 ± 0.00	**0.83** ± 0.01	**0.82** ± 0.01

Table 3.5: RMSE (on pIC50) on validation and test data (ligand split). For each model and data partition we mark the better representation in bold.

Data	Configuration	k-NN	BiMCA	BiMCA (pretrained)
Val.	Full seq.	**0.83** ± 0.01	0.75 ± 0.00	0.78 ± 0.01
	Active site	**0.83** ± 0.01	**0.79** ± 0.00	**0.80** ± 0.01
Test.	Full seq.	**0.83** ± 0.01	0.74 ± 0.00	0.77 ± 0.01
	Active site	**0.83** ± 0.01	**0.79** ± 0.00	**0.80** ± 0.01

Table 3.6: Pearson correlation coefficient on validation and test data. Same legend like Table 3.5.

nificant across the ten folds for both validation and test data as well as RMSE and pearson correlation as metrics ($p < 0.001$, W+).

However, the table also suggests that the k-NN model performs similarly on both active sites and full sequences. This observation is explained by the fact that the protein information is of negligible performance for our k-NN model *in a ligand split*. When retrieving the $k = 13$ nearest neighbors according to Equation 3.2, the first part (which measures protein similarity) will collapse to 0 for all samples of the same kinase. Note that this collapse occurs irrespective of the utilized kinase sequence and thus dilutes differences between the representations. As the average number of samples per protein in the dataset is 593 (see histogram in Figure 3.5**C**), it is not surprising that for the active site and full sequence indeed in 98.9% and 99.3% of the predicted samples, the nearest neighbor is a sample with the same kinase. To remedy the described confound and compare the impact of the two representations for the k-NN on the ligand split, we evaluated the performance exclusively on the remaining samples. For this small subset, the active site model is, in alignment with the overall findings, clearly superior to the full sequence model (RMSE 1.35 vs. 1.59, Pearson's r 0.56 vs. 0.33 on the test data). A remarkable side observation is that on those samples the active site BiMCA model surpasses its k-NN equivalent by a large margin (RMSE $= 1.18$, Pearson's $r = 0.64$). This indicates that the k-NN model strikes at interpolation, but falls behind the BiMCA in extrapolation; a hypothesis that is corroborated by an increased correlation of the prediction error with the distance to the nearest neighbor (k-NN: $r = 0.23$, BiMCA: $r = 0.18$; active site models, validation data).

Kinase inhibitor classes. We then sought to investigate the performance of the models for different groups of kinase inhibitors. To that end, we retrieved the primary target for each kinase inhibitor from BindingDB and grouped the ligands into thirteen groups, based on their alleged mechanism of action (MOA). To assign the primary targets to a potential MOA we used the classification scheme by *Roskoski Jr* [190]. From a total of around 372k validation samples, only about a third could be automatically assigned to a kinase inhibitor class. The PCC for each inhibitor class is shown in Figure 3.9 for both models and configurations. The plot shows that, with the exception of MEK inhibitors, the active site configuration yielded better results for all thirteen kinase inhibitor groups. While this is generally reassuring with respect to the overall hypothesis, let us assess why the performance for MEK (i.e., MAPK/ERK) inhibitors is consistently higher in full sequence models. Remember that the sequence alignment only relied on ATP binding site residues [185].

However, the class of MEK inhibitors includes several allosteric binders, i.e., ATP-noncompetitive inhibitors which bind to a unique site near but *outside* the ATP binding pocket [191]. In support of that, 94% of the 2909 MEK-inhibitor related samples making up this effect are indeed accounted for by eight kinases of the MAPK family (MKNK2, MKNK1, MAPKAP2, MAPK3, MAP2K1, MAPK1, MAPK14, MAP3K5).

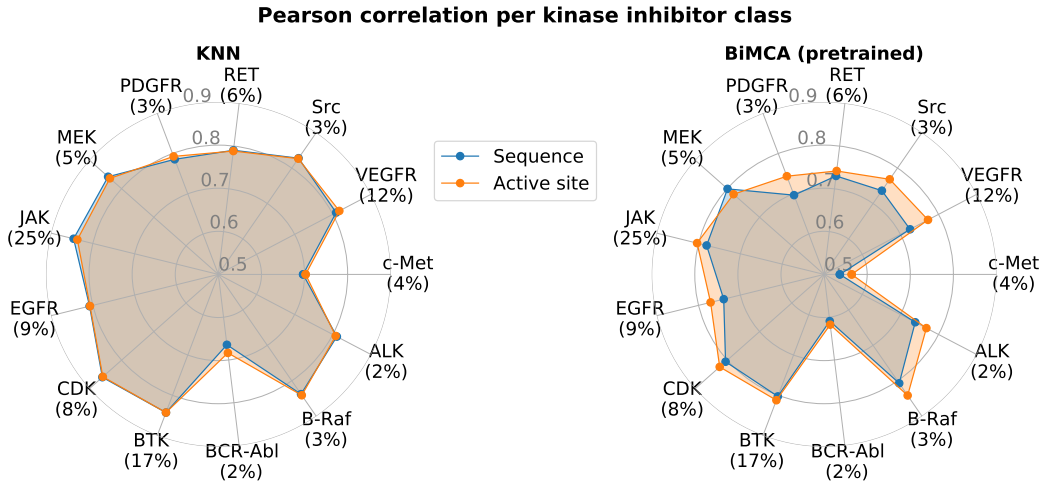


Figure 3.9: Performance in predicting affinity for novel kinase inhibitors, grouped by their alleged MOA. For the k-NN (left) and the pretrained BiMCA (right) the PCC of all samples of respective kinase inhibitor class is shown. Note that the differences in the k-NN are negligible due to the mechanism described above.

Similarity analysis. Binding affinity prediction models which generalize to distant manifolds of the chemical space are critical to successfully screening large virtual libraries. Similar to the kinase split, one might suspect that the performance on the ligand split depends on the availability of similar ligands during training. To rebut this hypothesis, Figure 3.10 shows that both models only exhibited a weak negative correlation between the per-ligand RMSE and the Tanimoto similarity to the nearest neighbor in training data (cf. Figure 3.10). Similar to the kinase split, the active site model does not only outperform the sequence model but is also less dependent on the availability of similar samples during training. The k-NN showed a slightly stronger negative correlation (PCC $= -0.23$, not shown).

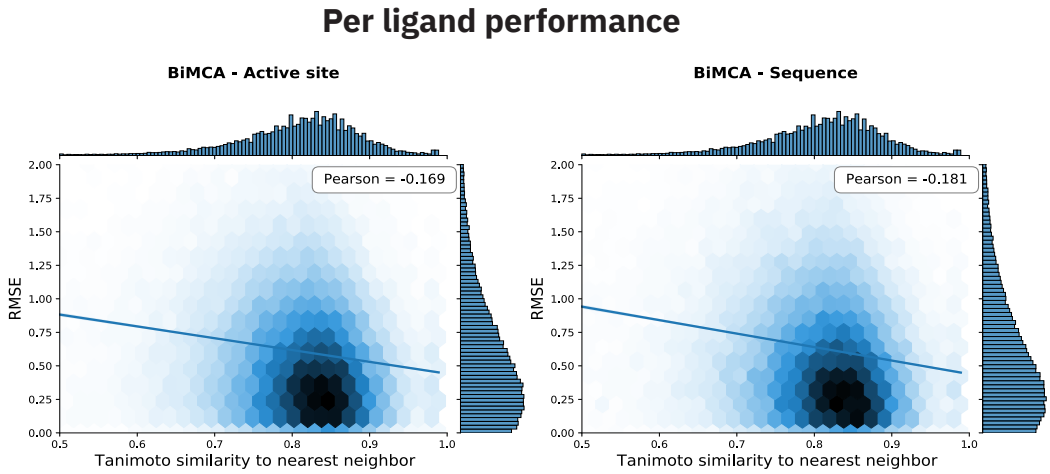


Figure 3.10: Dependency of prediction performance on availability of similar ligands. For each ligand, the (RMSE) is shown as a function of the Tanimoto similarity to the nearest training ligand. The colour gradient shows the density of the molecules and the line shows the correlation between both axes. Measures computed on validation data.

3.6.4.3 Ablation study on embedding types

To verify that the observed superiority of the active models could not be attributed to the amino acid embedding type, we compared the effect of our learned embeddings with one-hot encodings and the (standardized) BLOSUM62 matrix [192]. BLOSUM62 encodes amino acids based on the evolutionary similarity to all other amino acids.

From the results in Figure 3.11 we can see that, irrespective of the embedding type, the general trend regarding active site superiority manifests in both kinase and ligand split. While slight differences between the embedding types can be observed, e.g., the learned embeddings overall performed best, not even the best full sequence model reached the performance of the worst active site model (even disregarding the embedding type).

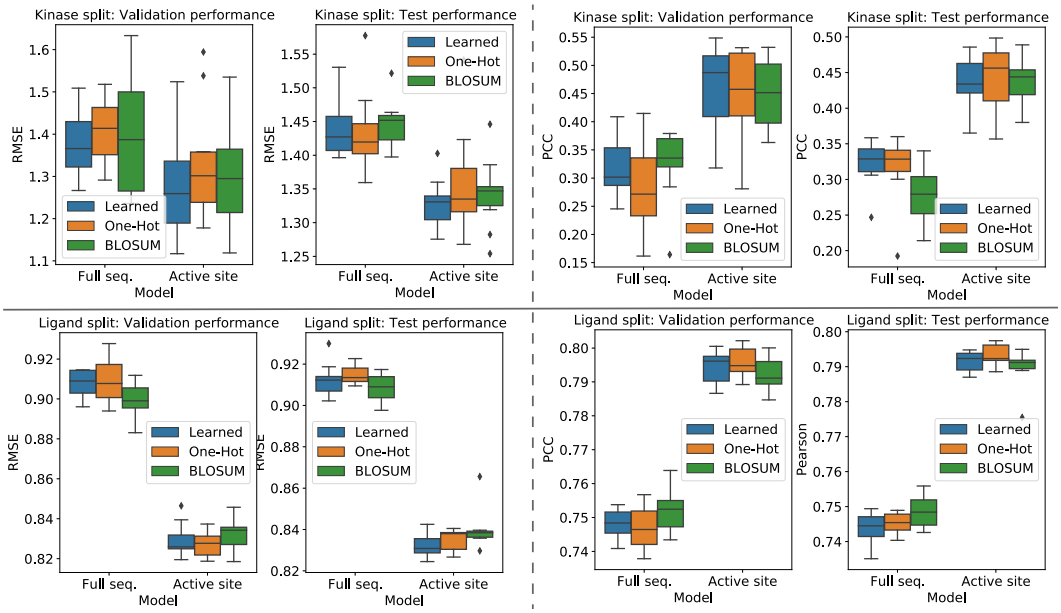


Figure 3.11: Ablation study on protein embeddings. Results for kinase and ligand split are shown in the top and bottom row respectively. The two left and two right columns show performance in terms of RMSE and PCC respectively; each compared across three protein embedding types. Exact numerical results can be found in Table A3.3.

3.6.4.4 Validation on external test dataset

To asses the robustness of our results, we verified the hypothesis on an independent test dataset. In particular we utilized the data from the IDG-DREAM challenge [146], released in 2021. The challenge focused on under-studied parts of the human kinome to catalogue the unexplored target space of kinase inhibitors. Thus, it resembles a particularly challenging dataset, encompassed by 825 samples (cf. Supplementary Data 1 by *Cichońska* et al. [146]). We shrank down the dataset to only include kinases for which full sequence and active site information [185] was available. This led to 720 samples, distributed across 276 kinases (32 unseen) and 93 ligands (all unseen). This dataset is much more than challenging both previously studied splits because:

1. for many samples *both* ligands and kinases are unseen.
2. experimental differences in the dose-response assays (multi-dose assays with maximal concentration of 10μM that cause an incorrect lower limit for activity)
3. the dose response metric, given in logarithmic dissociation constant (pK_d) that substantially differs from the pIC50 in BindingDB.

For the k-NN model we used all data available in BindingDB as training data whereas for the BiMCA we built an ensemble of the 10 models from the ligand split. The re-

Model	Config	All	Known kin.	Unknown kin.	Round 1	Round 2
k-NN	Full seq.	0.224	0.242	0.032	0.132	0.32
	Active site	0.244	0.282	-0.141	0.145	0.344
BiMCA	Full seq.	0.16	0.169	0.064	0.102	0.185
	Active site	**0.32**	**0.327**	**0.238**	**0.179**	**0.412**

Table 3.7: Evaluation on external dataset by *Cichońska* et al. [146]. We report Pearson correlation (PCC).

sults can be found in Table 3.7 and are confirming our overall findings. Again, in both models the active site residue representation outperforms the full sequence model. Also, the BiMCA again yields better results than the k-NN model. Notably, the active site BiMCA is the only model that achieves a satisfying performance in predicting activity in the under-studied kinases from *Cichońska* et al. [146] that were not included in BindingDB. We emphasize that a direct comparison to the results reported in the IDG-DREAM challenge is not possible due to the described differences.

3.6.4.5 Dissecting attention – why less is more

Model attention analysis. Theoretically, the full sequence BiMCA is strictly more expressive than their active-site BiMCA since they use a superset of residues and can exploit information from the entire protein. We propose two potential explanations for why they perform worse in practice.

1. First, the signal-to-noise ratio (SNR). Unlike the active site models, the full sequence models have to learn recognizing and disregarding residues that are largely irrelevant for binding.

2. Secondly, remember that the residues comprising the active site are contiguous in the folded protein but discontiguous in the full sequence (cf. Figure 3.4). The active site sequences thus carry implicit information about the 3D structure.

To investigate both hypotheses we examined whether the model learned to capture 3D information from the full sequence alone. This can be assessed from the attention weights of the BiMCA's context attention mechanism (cf. Equation 3.1).

If the full sequence model would have learned to focus on the active site residues, we would expect it to perform as well as the active site models.

Many previous publications [119, 120, 157] provided visualizations of amino-acid-level attention and argued with case studies that the attention mechanism can capture protein interaction sites. However, these analyses were of qualitative nature and it was later demonstrated in a rigorous quantitative evaluation that none of those models systematically highlights interaction sites [121]. Instead, *Li* et al. [121] showed that explicit super-

vision is required to excel at predicting pairwise non-covalent interactions. For two exemplary kinases, MAPK11 and ABL1 we performed an interpretability analysis to assess whether the sequence model paid significantly higher attention to the active site residues (see Figure 3.12). Each protein-ligand pair ($N = 39$ for MAPK11 and $N = 749$ for ABL1) is shown as one point in Figure 3.12. Following the methodology by *Li* et al. [121] for interaction site prediction, we measured the model's ability to highlight the active site by two metrics. First, the AUC between the binary labels and the per-residue attention scores. Secondly, the *enrichment score*; a precision-based metric derived from the binarized attention values. The enrichment score accounts for sequence length and randomly expected hits. Both metrics were computed per sample and averaged across all samples of a protein Thirdly, we evaluated statistical significance with a one-sided Mann-Whitney-U test.

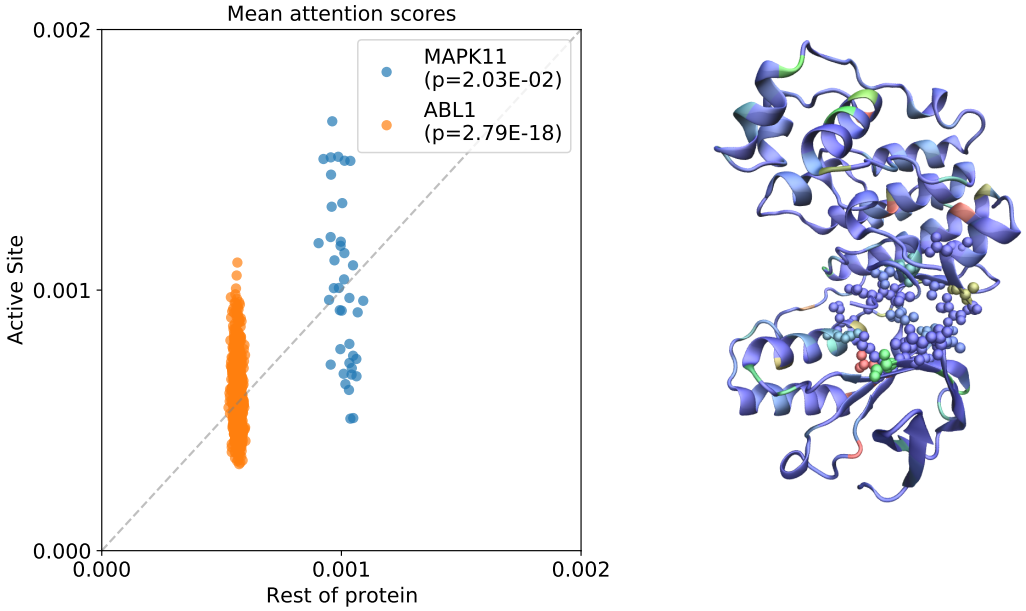


Figure 3.12: Residue-level attention scores. *Left*: For each kinase-ligand pair of MAPK11 and ABL1, the mean attention scores on active site residues versus the remaining residues is shown. If the model would assign equal attention to all residues, all points would be on the dashed line. *Right*: Attention heatmap on the MAPK11 3D structure highlighting atoms with high attention scores (blue means low, green medium and red high attention). Residues depicted as spheres belong to the active site.

The AUC scores for MAPK11 and ABL1 are 0.518 and 0.516 respectively (AUC of random classifier: 0.5). The average enrichment scores are 1.16 and 3.48 (random classifier: 1). In the comparison by *Li* et al. [121], all investigated unsupervised attention-based methods [119, 120, 157] achieved AUCs around 0.5 and enrichment scores around

1 whereas explicit supervision on the interaction site yielded an AUC of 0.76 and an enrichment score of 10.7. Exact comparison, however, is not possible because the analysis was performed on different samples and our notion of an active sites differs from their definition of an interaction site. We also find that for both kinases, the mean attention scores on the active site residues are significantly higher than on the remaining residues ($\alpha < 0.05, MWU$).

Interpretation. These results are insightful to understand why the active site models performed better. Confirming the results of *Li* et al. [121], the BiMCA (just like all previous attention-based methods in binding affinity prediction) does not convincingly predict the active/interaction site when supervision only occurs on binding affinity labels. However, the BiMCA shows a mild but significant ability to focus on relevant residues. While this trend is not consistent across all samples (cf. Figure 3.12), all three quantitative scores (AUC, enrichment and MWU test) suggest that the BiMCA performs significantly above chance level in extracting active sites.

Conclusively, we believe that these subtle 3D effects in the full sequence model are falling much behind the 3D information conveyed in the active site models. Together with the higher SNR this could contribute to their improved generalizability compared to full sequence models.

3.7 On the choice of active site sequences

3.7.1 Extending the active site definition

In the previous section, we found that the superiority of active site representations manifests consistently for all but one ligand type – namely: MEK/MAPK inhibitors (cf. Figure 3.9). This outlier is not surprising given that this class contains numerous allosteric binders, in particular ATP-noncompetitive MAPK inhibitors [191]. In this section, we tackle this limitation in modeling allosteric binders and refine the definition of an "active site" for binding affinity prediction. To achieve that, we utilize an alternative active site definition comprising 16 residues from *Martin* and *Mukherjee* [138] which includes 6 residues farther away from the immediate binding site (see Figure 3.13**A**). Those *Martin* residues were identified with a variable selection algorithm from a starting set of 46 residues based on how frequently they were picked for a large set of kinase-kernel models. Since only 10 of these 16 residues are overlapping with the *Sheridan* definition, we also examine a *Combined* active site definition with 35 residues (cf. Figure 3.13**B**).

Definition 3.7.1 (*Martin* active site [138])**.** The 16 residues comprising the *Martin* active site are defined as follows: $\mathcal{R}_{\text{Martin}} = \{$ D127, E121, F187, F54, G126, I163, L103, L106, L162, L173, L95, M120, T183, T51, V119, V123 $\}$.

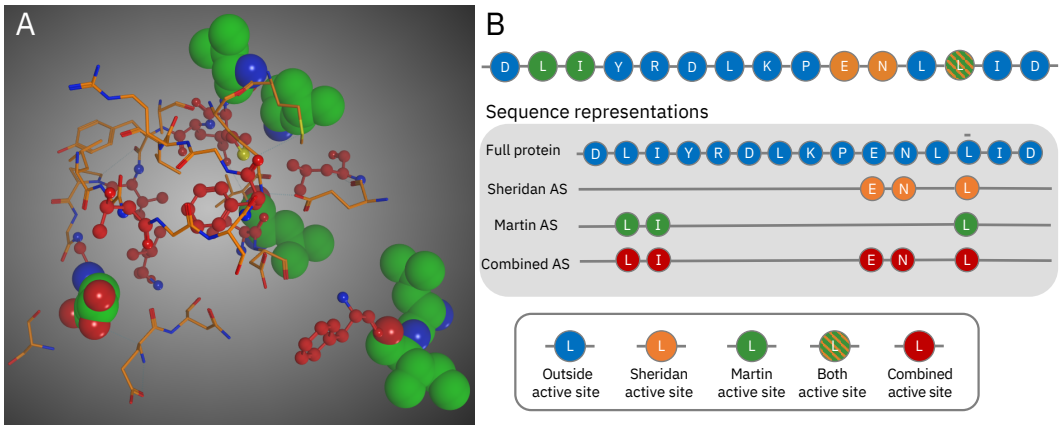


Figure 3.13: Overview of sequences obtained with different active site site definitions. **A)** Visualization of cAMP-dependent protein kinase catalytic subunit alpha (P17612). Residues unique to the active site definitions of *Sheridan* et al. [183] and *Martin* and *Mukherjee* [138] are shown in orange and green. Residues contained in both definitions are colored in red. **B)** Partial amino acid sequence (residues 48-62) of the same kinase. Below we show the three different active site representations.

The 10 underlined residues are shared between definition the *Sheridan* and the *Martin* definition.

Definition 3.7.2 (*Combined* active site)**.** The 35 residues comprising the *Combined* active site are: $\mathcal{R}_{\text{Sheridan}} \bigcup \mathcal{R}_{\text{Martin}}$.

3.7.2 Extended performance comparison

Ligand split. The results in Table 3.8 confirm the superiority of using active sites rather than full sequences, irrespective of the exact definition of active site. Importantly, the results also clearly demonstrate that the *Combined* representation yields consistently the best results for both models, both metrics and validation and test data. These improvements are statistically significant ($W+$) compared to at least one active site definition for all settings (see Figure 3.14 and Figure A3.1). Several observations can be made:

1. Importantly, the inferiority of the *Sheridan* active site definition for MEK inhibitors prediction can be resolved using the *Martin* or the *Combined* active site definition which includes 6 more distant residues (cf. Figure 3.14**C**). Definition 3.7.1 includes residues around the "hydrophobic spine", which presumably affect the stability of binding site features or the active and inactive forms [193].

2. The *Martin* definition also includes T51, a residue that builds an important salt bridge with residues in the same loop in many CDK kinases, another class where *Martin* and *Combined* are better than *Sheridan*.

Data	Config.	RMSE (↓)		Pearson (↑)	
		BiMCA	BiMCA-pre	BiMCA	BiMCA-pre
Val.	Full sequence	$0.908_{\pm 0.01}$	$0.848_{\pm 0.01}$	$0.748_{\pm 0.00}$	$0.782_{\pm 0.01}$
	AS (Sheridan)	$0.829_{\pm 0.01}$	$0.821_{\pm 0.01}$	$0.794_{\pm 0.00}$	$0.797_{\pm 0.01}$
	AS (Martin)	$0.839_{\pm 0.01}$	$0.813_{\pm 0.01}$	$0.791_{\pm 0.00}$	$\mathbf{0.804}_{\pm 0.01}$
	AS (Combined)	$\mathbf{0.828}_{\pm 0.01}$	$\mathbf{0.811}_{\pm 0.01}$	$\mathbf{0.797}_{\pm 0.01}$	$\mathbf{0.804}_{\pm 0.01}$
Test	Full sequence	$0.912_{\pm 0.01}$	$0.863_{\pm 0.01}$	$0.744_{\pm 0.00}$	$0.774_{\pm 0.01}$
	AS (Sheridan)	$\mathbf{0.832}_{\pm 0.01}$	$0.826_{\pm 0.01}$	$0.792_{\pm 0.01}$	$0.795_{\pm 0.01}$
	AS (Martin)	$0.842_{\pm 0.01}$	$0.818_{\pm 0.01}$	$0.789_{\pm 0.01}$	$0.801_{\pm 0.01}$
	AS (Combined)	$\mathbf{0.832}_{\pm 0.01}$	$\mathbf{0.816}_{\pm 0.01}$	$\mathbf{0.795}_{\pm 0.01}$	$\mathbf{0.802}_{\pm 0.01}$

Table 3.8: Results on validation and test data (ligand split). 10-fold cross validation results on kinase data from BindingDB. For each model and data partition we show mean and standard deviation across 10 folds and mark the best representation in bold.

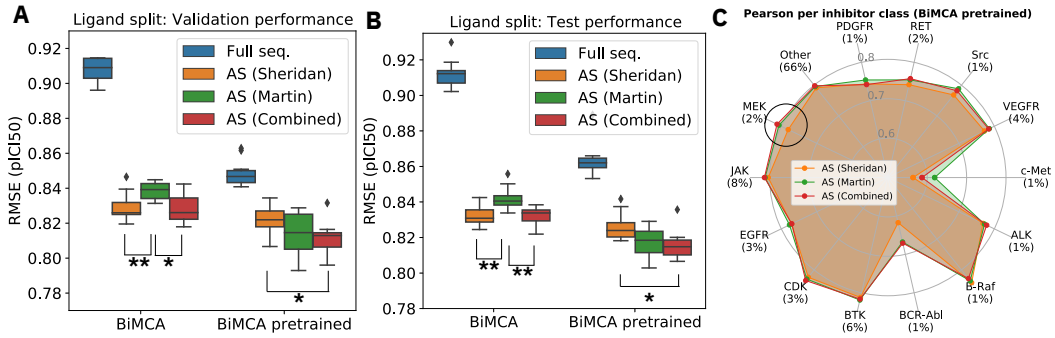


Figure 3.14: Ligand split performance. The RMSE in pIC50 prediction for four kinase representations and two models on validation and test data is shown respectively in **A)** and **B)**. Statistically significant differences between the three different active-site configurations are marked with stars dependent on their significance level. **C)** Performance comparison across representations as grouped by kinase inhibitor class. For details see text.

3. Thirdly, the *Sheridan* definition is surprisingly inferior to the other two despite most BCR-Abl binders being ATP-competitive [194]. Note however, that performance is generally poor, most likely caused by much less training data per sample, compared to other classes.

KINASE SPLIT. The results for the k-NN and the BiMCA on the validation and test data are shown in Figure 3.15**A** and **B** respectively. No clear trend can be seen on the validation data when comparing the three active site configurations across models, data splits and metrics. While the *Sheridan* representation is significantly superior to the *Martin* representation for the k-NN ($p < 0.05$, $W+$) and to the *Combined* representation for the BiMCA, this trend does not persist in the test data. During testing, the *Martin*

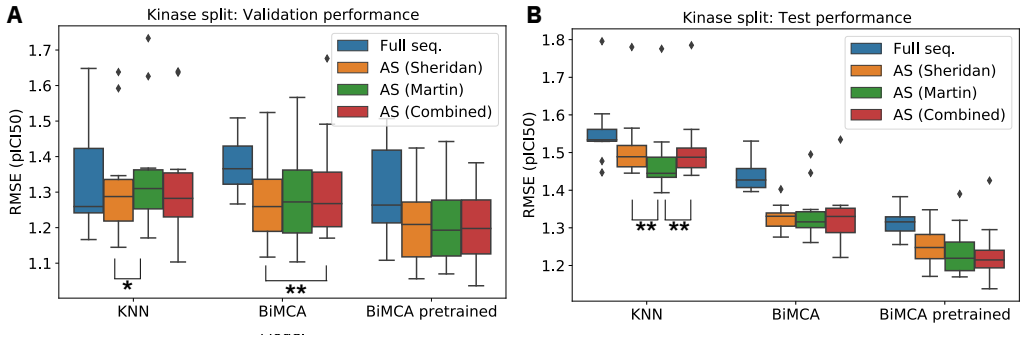


Figure 3.15: RMSE in affinity prediction for kinase split on validation and test data. 10-fold cross validation results on kinase data from BindingDB. Performance on validation (**A**) and test data (**B**) are shown. Statistically significant differences between the three different active-site configurations are marked with a star.

representation consistently obtained the highest Pearson correlation, irrespective of the model (cf. Table A3.4). Notably, our best model (the pretrained BiMCA) obtained the best performance with the *Combined* representation in all but one cases.

3.8 Active site sequence augmentation

3.8.1 Introducing novel augmentation strategies

To conclude the experiments of this chapter, we explore additional mechanisms to leverage the knowledge about the location of the active site, in particular how it can inspire data augmentation. We propose two new protein sequence augmentation techniques and find that they have complementary, positive effects.

Given that the MSA-extracted sequence does not provide explicit 3D information, proximity in the sequence is likely but not certainly corresponding to proximity in 3D space. We thus conjectured that sequence augmentation strategies could assist to learn general binding patterns for two reasons:

1. There may be 1D representations that align better with the 3D relation of residues than the original sequence. Representing a kinase as a distribution of sequences reflects this lack of knowledge, might regularize the model and thus improve generalization, especially to unseen target families.

2. Static roles of specific residue positions may induce overfitting in practice as the model might memorize too specific patterns.

As shown in Figure 3.16, we therefore devise three novel sequence augmentation schemes, two of which exploited the discontiguity of the active site residues in the full protein:

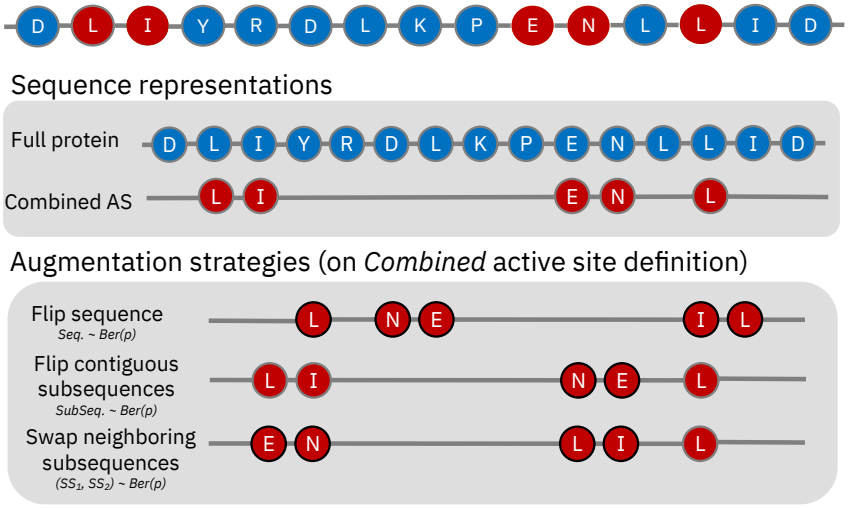


Figure 3.16: Protein sequence augmentation. Three possible kinase sequence augmentation strategies, exemplified on the *Combined* active site definition: 1) flipping (i.e., reversing) the entire sequence; 2) randomly flipping contiguous subsequences and; 3) randomly swapping neighboring subsequences. Residues affected of the augmentation are encircled in black.

- **Flipping (F)**: A natural augmentation technique for protein sequences is flipping the entire reduced residue set (applied with a probability of 50%).
- **Flip contiguous subsequences (FS)**: Contiguous subsequences of the active site are closely together in space. Similar to **Flipping**, this strategy relies on the fact that reading such sequences from either direction should not affect model predictions (applied with 50% probability).
- **Swap neighboring contiguous subsequences (SS)**: This strategy relies on the assumption that neighboring contiguous sequences have a higher probability to be closer in space than distant active site subsequences (20%).

While the second and the third strategy revoke the residue ordering from the MSA, we hope that this biologically constrained rearrangement eases learning patterns of binding.

3.8.2 Performance comparison

All experiments rely on the *Combined* representation because it previously yielded the best results. The results in Table 3.9 demonstrate that all augmentation techniques improved model performance. Note that the structure-motivated augmentation methods like swapping (SS) and flipping subsequences (FS) showed a similar performance boost to plain flipping (F). But while the benefit of flipping is statistically insignificant, the FS

Data	Augmentation	RMSE (↓) BiMCA	RMSE (↓) BiMCA-pre	Pearson (↑) BiMCA	Pearson (↑) BiMCA-pre
Val.	None	$1.32_{\pm 0.16}$	$1.20_{\pm 0.12}$	$0.438_{\pm 0.08}$	$0.489_{\pm 0.09}$
	Flip (F)	$1.25_{\pm 0.13}$	$1.19_{\pm 0.13}$	$0.463_{\pm 0.08}$	$0.502_{\pm 0.08}$
	Flip sub-seq. (FS)	$1.28_{\pm 0.12}$	$\mathbf{1.18}_{\pm 0.12}$	$0.431_{\pm 0.11}$	$\mathbf{0.521}_{\pm 0.08}$
	Swap sub-seq. (SS)	$1.28_{\pm 0.17}$	$\mathbf{1.18}_{\pm 0.12}$	$0.443_{\pm 0.11}$	$0.511_{\pm 0.09}$
	FS + SS	$1.27_{\pm 0.11}$	$\mathbf{1.18}_{\pm 0.12}$	$0.444_{\pm 0.09}$	$0.508_{\pm 0.09}$
	F + FS + SS	$\mathbf{1.22}_{\pm 0.10}$	$\mathbf{1.18}_{\pm 0.11}$	$\mathbf{0.468}_{\pm 0.11}$	$0.505_{\pm 0.09}$
Test	None	$1.33_{\pm 0.08}$	$1.23_{\pm 0.08}$	$0.431_{\pm 0.06}$	$0.505_{\pm 0.07}$
	Flip (F)	$1.28_{\pm 0.05}$	$1.23_{\pm 0.07}$	$0.478_{\pm 0.04}$	$0.515_{\pm 0.06}$
	Flip sub-seq. (FS)	$1.32_{\pm 0.09}$	$1.22_{\pm 0.04}$	$0.444_{\pm 0.08}$	$0.516_{\pm 0.04}$
	Swap sub-seq. (SS)	$1.28_{\pm 0.04}$	$1.23_{\pm 0.03}$	$\mathbf{0.479}_{\pm 0.01}$	$0.506_{\pm 0.06}$
	FS + SS	$1.29_{\pm 0.06}$	$1.22_{\pm 0.07}$	$0.469_{\pm 0.04}$	$0.526_{\pm 0.05}$
	F + FS + SS	$\mathbf{1.27}_{\pm 0.06}$	$\mathbf{1.21}_{\pm 0.05}$	$\mathbf{0.479}_{\pm 0.06}$	$\mathbf{0.531}_{\pm 0.05}$

Table 3.9: Results of sequence augmentation (kinase split).

and SS configuration yield significant benefits ($p < 0.01$, W+) in several cases. Interestingly, their performance increase is roughly additive. Upon combining all augmentation strategies we obtain the best results in seven out of eight cases ($p < 0.01$, W+, RMSE on validation data). Another interesting aspect is that the pretrained model is harder to improve; presumably because it may have learned to be invariant against the applied transformations.

3.9 Discussion

The experiments on proteochemometric language modeling of human kinases in this section suggest a superiority of active site to full protein sequences for binding affinity prediction. This finding is robust across two investigated models (a k-NN regressor and a proteochemometric language model), different data splits (kinase and ligand split) and performance metrics (RMSE and PCC) and were confirmed on the largest existing dataset (BindingDB) as well as a new, external test dataset [146].

This is an important, and maybe surprising finding because the active site definitions contain only a tiny subset of the residues in the full primary sequence. We believe that providing exclusively the active site residues increases the SNR and implicitly conveys information about the 3D structure which consequently yields better performance.

This hypothesis is partly corroborated by our attention analysis. Without explicit supervision on residue importance the sequence model learns only to a small extent to focus on the relevant, active site residues. In contrast to this scarce signal of tertiary structure in

the full sequence models, the active site models are equipped with an inductive bias about 3D structure provided by the discontiguity of the active site residues in the full sequence.

Moreover, we find that the active site models even surpass the full sequence models if both models were pretrained on full protein sequences. This result suggests that it is beneficial to pretrain proteochemometric models on pan-protein data even if the ultimate user-application is limited to a specific protein family.

Since our initially utilized *Sheridan* active site did not yield satisfying results on predicting binding affinity for ATP-noncompetitive inhibitors, we conducted additional experiments using an active site definition proposed by *Martin* and *Mukherjee* [138]. Taking the union of both definitions, we find that our novel *Combined* kinase representation is superior to the *Sheridan* as well as the *Martin* active site definition for binding affinity prediction for unseen ligands. Notably, this *Martin* definition included residues distant from the ATP-binding site which improved performance for allosteric binders. Some residues in the "hydrophobic spine" might take dynamical roles [195]. Other residues such as G126, I16, T51, L103 and V119 are relevant for activation-deactivation mechanism and loop dynamics, but do not directly interact with the ligand [138].

Finally, we exploited the knowledge about active site residue location to devise several novel sequence augmentation techniques. In our experiments, they exhibited further and complementary performance improvement. In sum, our results improve the efficient modeling of kinase-ligand binding.

Future research could validate our findings on other protein families or explore hybrid approaches to leverage 3D information. For example, one could constrain the attention mask to reflect pairwise non-covalent compound-protein residue interactions with semi-supervised learning. While active site information might not be available for some protein families, existing methods could be applied to extract specific protein-ligand binding residues [196].

Part II

Conditional molecular design

4 Conditional molecular generative models

4.1 Molecular generative modeling landscape

In the past few years, deep molecular generative models are steadily growing their influence in computational chemistry and start to impact industrial molecular discovery. In the last years, we have witnessed a paradigm shift away from discrete, local optimization toward a systematic chemical space exploration [10]. In a seminal work, *Gomez-Bombarelli* et al. [11] introduced a SMILES-based VAE that was able to embed molecules into a smooth latent space which facilitated not only sampling of novel molecules but also a meaningful chemical space interpolation. Almost concurrently, *Segler* et al. [55] demonstrated that the molecules generated through a RNN mimicked the distributions of physicochemical properties from the training data and could even be tuned toward specific properties of interest. A popular optimization technique to generate molecules with desired properties is reinforcement learning (RL); often coupled with RNNs by treating the SMILES generation as action sequence and the molecular property as reward. Such approaches can utilize model-free [197] and model-based RL techniques [198]. A seminal work by *Popova* et al. [18] provided evidence that RL methods like policy gradients can steer RNN-based SMILES generators toward desirable properties like synthesizability, JAK-inhibition or solubility [18]. RL can also be readily coupled with adversarial techniques to obtain stochastic policies for molecular property optimization, as shown in ORGAN [199].

Graph-based molecular generative models are also getting more popular. These methods either generate graphs in one-shot or auto-regressively. One-shot generation have the disadvantage that the validity of the molecular graphs is difficult to guarantee [200, 201]. Auto-regressive molecular graph generative models either generate one node [25, 202], sets of nodes [203] or edges [204] at a time. In the Junction-Tree VAE proposed by *Jin* et al. [19], functional groups are generated in a tree-structured manner and then combined via message passing to guarantee valid molecular graphs. *Shi* et al. [24] developed an auto-regressive graph generative model that relies on normalizing flows [205] and achieved superior performance on property optimization benchmarks. Recently, *Bengio* et al. [206] proposed GFlowNets, an active learning method that was, among others, demonstrated

to improve on diverse candidate generation [207]. Diffusion models have also recently been proposed for molecular conformer generation [208, 209].

As discussed in the Introduction (Section 1.2) most of these models suffer from two problems. First, they are task-specific and can thus only be applied to exactly *one* objective and secondly they disregard system-level information about the environment in which the molecule has to exert its function.

Throughout this chapter, we will strive to build more flexible generative models that can be queried with a "semantic context" and can generate molecules for a wide range of desired conditions without the need of specific optimization. As visualized in Figure 1.2, we will formulate and apply conditional molecular generative models for four types of contexts:

1. **Cell profiles**: In this task, the objective is to develop a model that can be conditioned on an omic profile (e.g., gene expression data from a malignant tumour) and generate molecules that are likely to exhibit a high inhibitory effect against the cell profile. This will be described in Section 4.3.

2. **Protein targets**: In this task, the objective is to develop a model that can be conditioned on a protein sequence and generate molecules (i.e., ligands) that are likely to bind to the protein. This will be described in Section 4.4.

3. **Molecular substructures** (e.g., scaffolds): In this task, the objective is to develop a model that can be conditioned on a seed molecule (or even an explicit, possibly discontiguous molecular substructure) and generate molecules that are 1) similar to the seed *and* 2) exhibit desired properties. Together with **4.** this will be assessed in Chapter 5.

4. **Continuous properties** (e.g., a desired solubility value): In this task, the objective is to develop a model that can be conditioned on a desired floating-point property value and generate molecules adhering to the property constraint. Together with **3.** this will be assessed in Chapter 5.

For each task, the related work will be discussed in the respective section/chapter.

4.2 PaccMann^{RL}: Coupling a hybrid VAE to property predictors via RL

In this section we introduce `PaccMann`^{RL}, a hybrid Variational Autoencoder (VAE) for conditional molecular generation that is trained with RL through a multimodal property prediction model. The PaccMann^{RL} framework can be applied to different settings in molecular discovery. As visualized in Figure 4.1, three key ingredients are necessary to form a PaccMann^{RL} model:

- **Molecular VAE**: This is an unsupervised model that is able to encode and generate molecules. It will be denoted by Θ_M.
- **Context VAE**: This is an unsupervised model that is able to encode and generate context instances. The choice of the *context* determines the application usecase of the specific PaccMann$^{\text{RL}}$ instance. It will be denoted by Θ_C. Together with Θ_M this will form the "agent" G_Θ.
- **Multimodal property prediction model**: This model receives a molecule and a sample from the context VAE and predicts an interaction effect between the two entities. It will be denoted by R_I and be the main part that composes the "critic".

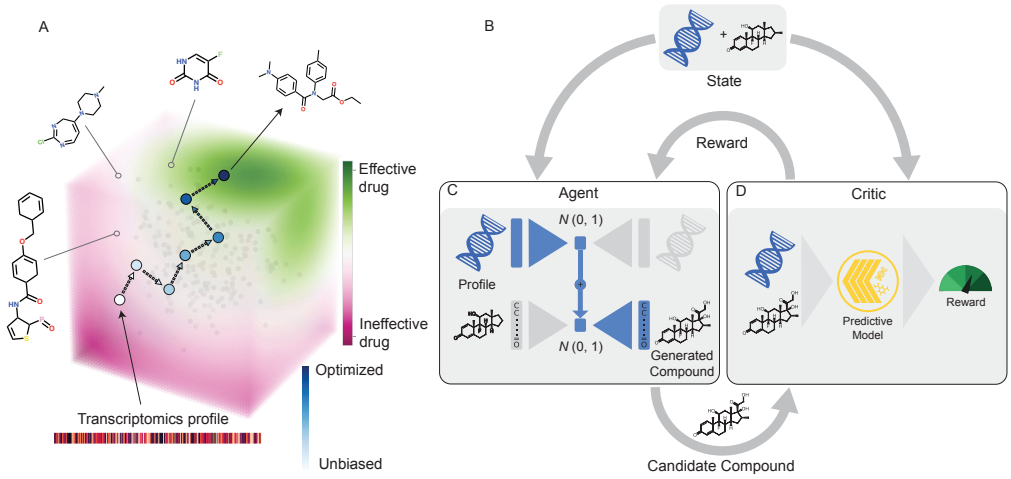


Figure 4.1: The PaccMann$^{\text{RL}}$ framework for conditional molecular design. Exemplified on the task of cancer-profile driven molecular generation. **A)** Conceptual depiction of model training process. Starting from an unbiased molecular generation, we learn to navigate through the chemical space toward a manifold that is more densely populated with molecules that give a higher reward (*here*: predicted inhibition of desired cancer profiles). **B)** PaccMann$^{\text{RL}}$ model. The conditional molecular generator is embodied through a pretrained hybrid-VAE. The generative process starts with encoding a transcriptomics profile through a pretrained omics VAE. The obtained latent code of the profile is then passed to a separately pretrained molecular decoder (see **C**). Next, the generated molecule is evaluated by a "critic" (*here*: a multimodal drug sensitivity prediction model which consumes the molecule as well as the profile (see **D**). We close the loop by interpreting the predicted efficacy as reward and optimize the agent with RL to generate molecules that produce a higher reward.

All three models have to be pretrained independently. Two applications of the PaccMann$^{\text{RL}}$ model are explored in this thesis; namely using transcriptomics profiles of cancer cells and protein sequences as *context*. Further possible applications include

generation of molecules that inhibit cell profiles represented by other types of bulk or single-cell omics data, have high synergistic effect with a target drug or even induce a desired inhibition profiles. We believe that VAEs are the ideal model for this task since they learn a structured latent space which facilitates the combination of different modalities.

4.2.1 Problem formulation

Let $m \in \mathcal{M}$ denote a molecule from the molecular space and $c \in \mathcal{C}$ be an (abstract) context[1].

Objective 4.2.1. Our goal is to learn a mapping $G_\Theta : \mathcal{C} \to \mathcal{M}$ subject to maximization of the function $\Phi(c, m)$.

4.2.1.1 Critic

$\Phi(\cdot, \cdot)$ is a multimodal reward, typically an interaction effect between c and m (e.g., the binding strength between a protein and a molecule). Since the exact computation of Φ is intractable (it requires an *in vitro* experiment), it is approximated with $R : \mathcal{C} \times \mathcal{M} \to \mathbb{R}$. Typically, $R := f \circ R_I$ where we call R_I the "critic" and $f : \mathbb{R} \to \mathbb{R}$ is an optional transformation converting the predicted interaction into a reward which will later be subject to maximization. R_I denotes the actual property predictor, e.g., a proteochemometric binding affinity prediction model. While R_I has to be pretrained independently, it will be fixed throughout this part of the thesis and thus considered an oracle. Different types of models that could constitute R_I were developed in the first part of this thesis.

4.2.1.2 Agent

Note that since G_Θ will be optimized using a RL scheme that relies on the critic above, we call G_Θ the "agent". Before defining G_Θ, let us first introduce two separate VAEs [210].

Definition 4.2.1. Let $\Theta_{\text{M}} = [\Theta_{\text{M}}^{\text{Dec}} \circ \Theta_{\text{M}}^{\text{Enc}}] : \mathcal{M} \to Z_M \to \mathcal{M}$ be a molecular VAE that is trained on $\mathcal{T}_{\text{M}} = \{m_i\}_{i=1}^{N_{\text{M}}}$.

Definition 4.2.2. Let $\Theta_{\text{C}} = [\Theta_{\text{C}}^{\text{Dec}} \circ \Theta_{\text{C}}^{\text{Enc}}] : \mathcal{C} \to Z_C \to \mathcal{C}$ be a context VAE that is trained on $\mathcal{T}_{\text{C}} = \{c_i\}_{i=1}^{N_{\text{C}}}$.

Θ_M and Θ_C are trained with unsupervised learning on molecular and context samples respectively. The objective of both VAEs is to optimize their ELBO:

$$\mathcal{L}_{\text{VAE}} := \mathbb{E}[\log p(\mathbf{x}|\mathbf{z})] - \mathcal{D}_{KL}[q(\mathbf{z}|\mathbf{x}), p(\mathbf{z})] \tag{4.1}$$

[1] this could e.g., be a protein from the protein space but we keep the formulation generic.

where $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\vec{0}, \mathbf{I})$, i.e., the latent code is modelled using a multivariate unit Gaussian following standard VAE convention. For detailed formulation of VAEs see *Kingma* and *Welling* [210] and *Sohn* et al. [211]. The only additional constraint is that $|Z_M| = |Z_C|$, i.e., both VAEs have the same latent code dimensionality.

Baseline. After training Θ_M, we can sample from $p(\mathbf{z}_M)$ and apply $\Theta_{\text{M}}^{\text{Dec}} : Z_M \rightarrow \mathcal{M}$. This process constitutes our baseline for molecular generation.

Conditional generation. After Θ_M and Θ_C are trained, our "agent", the conditional generative model G_Θ is defined as:

$$G_\Theta = [\Theta_{\text{M}}^{\text{Dec}} \circ \Theta_{\text{C}}^{\text{Enc}}] : \mathcal{C} \rightarrow Z \rightarrow \mathcal{M} \tag{4.2}$$

In explanation, our conditional generator is obtained by encoding a context sample c with the context VAE into its latent space before decoding this latent code with the molecular decoder. This mixture can be performed due to the variational constraint in both the context and the molecular VAE. Critically, this constraint in Equation 4.1 drives both models to encode their respective samples (i.e. context and molecules) into a multivariate Gaussian distribution with *the same* number of dimensions. Thus, the combination of the two models enables to learn a latent space that links the context space with the molecular space thus providing a mechanism to sample novel compounds given a context. During the optimization, this fusion will warp the latent space from encoding structural similarity into *functional* similarity that clusters molecules with a similar predicted reward, given a context c[2]. The final training objective function of the hybrid VAE G_Θ is to learn a policy $\Pi(\Theta)$:

$$\Pi(\Theta) = \sum_{m \in \mathcal{M}} P_\Theta(m|c) R(m, c) \tag{4.3}$$

where $P_\Theta(m|c)$ indicates the conditional probability approximated by G_Θ. In layman's terms we are trying to maximize the conditional probability of sampling a molecule m from a context c that gives a maximal reward $R(m, c)$. Since Equation 4.3 is intractable to compute, it is approximated using policy gradient and subject to maximization using REINFORCE [212], as proposed in ReLeaSE [18]. Critically, the performance of G_Θ is limited by the quality of the reward function R. Therefore, we have devoted the first part of this thesis to developing robust molecular property prediction models.

[2] A similar procedure was shown by *Gomez-Bombarelli* et al. [11] where a property predictor was trained on the latent space.

4.2.2 Molecular decoder

In our implementation the molecular decoder Θ_M^{Dec} is an auto-regressive (recurrent) network that builds a SMILES (or SELFIES) sequence token by token. In the following we describe the concept of context-driven molecular generation more closely. Let us assume that the first step of G_Θ is complete (i.e., the context c was embedded into a latent code $\mathbf{z}_c$ using Θ_C^{Enc}. At every time t of auto-regressive molecular generation, the state s_t is given by a tuple $(m_t, \mathbf{z}_c)$, i.e., a partial molecular string m_t and the embedded context vector $\mathbf{z}_c$. The agent then has to chose an action a_t from the action space $\mathcal{A}$ which is the vocabulary of all tokens of the chemical language. Formally:

$$p(a_t|s_{t-1}) \text{ where } s_{t-1} = (m_{t-1}, \mathbf{z}_c) \tag{4.4}$$

m_0 is simply the <START> token.

Note that Equation 4.3 assumed that the conditional probability $P_\Theta(m|c)$ approximated by G_Θ always results in valid molecules $m \in \mathcal{M}$. In practice, our implementation relies on chemical languages such as SMILES which can produce invalid molecules (i.e., strings $m_T \notin \mathcal{M}$). Hence it is more appropriate to reformulate Equation 4.3 to:

$$\Pi(\Theta) = \sum_{m_T \in \mathcal{M}^*} P_\Theta(m_T) R(m_T, c) \tag{4.5}$$

where

$$P_\Theta(m_T) := \prod_{t=0}^{T} p(a_t|s_{t-1}) \tag{4.6}$$

A terminal state $m_T \in \mathcal{M}^* \supset \mathcal{M}$ is reached when either $t = T$ or when the terminal action $a_t =$ <END> has been sampled (in which case we set $T := t$). The generative process is visualized in Figure 4.2**C**. To decode a SMILES string, we apply, at any time t, a multinomial sampling over the predicted (softmax) distribution over the SMILES vocabulary token. Unless stated otherwise we use a temperature parameter of 1. Note, that in our framework, the reward r_t is calculated only for the terminal states, $\mathcal{M}^*$, as previously defined. For all intermediate steps $t < T$, we set $r_t = 0$ (intermediate SMILES strings m_t are usually not valid anyways). Similarly, $R(c, m) = 0$ if $m \notin \mathcal{M}$ (i.e., final but invalid SMILES strings also receive no reward).

These choices imply that our formulation of the hybrid VAE is generic and thus, in principle the molecular decoder does not have to be sequence-based and/or autoregressive but could be adapted to produce graphs or fingerprints dependent on the desired molecular representation.

Learning to count. The SMILES language is a context-free language according to the Chomsky hierarchy [213]. Unlike natural languages, it requires a balanced set of

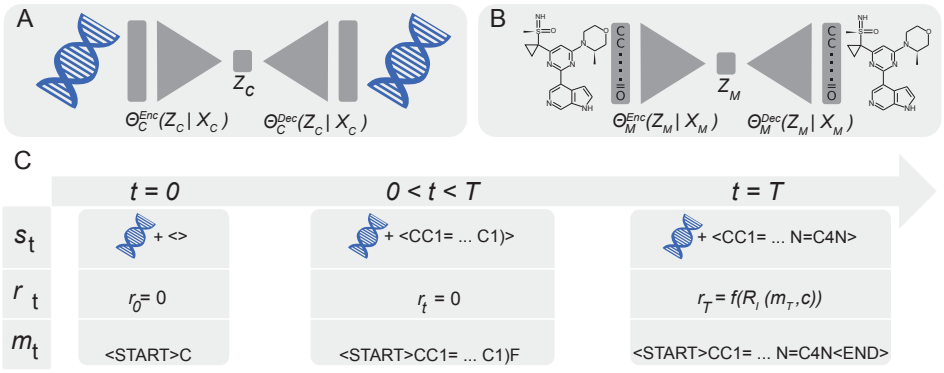


Figure 4.2: Construction of PaccMann^{RL} components. A) A context VAE, in this case trained on biomolecular profiles (RNA-Seq data). **B)** A molecular VAE with an autoregressive decoder pretrained on SMILES or SELFIES strings. **C)** The generative process in Θ_M^{Dec} during RL optimization. Molecules are generated auto-regressively as SMILES sequences. A full cycle of this process includes a state (s_t, where $s_0 = \mathbf{z_c}$, i.e., the latent code of the context vector), an intermediate reward (r_t) and a generated partial candidate molecule (m_t).

parentheses which poses an additional syntactical challenge. Hence, models that generate SMILES sequences auto-regressively benefit from an ability to *count* the branching tokens (the parentheses "`(`" and "`)`") as well as ring symbols (`1`, `2` etc.) in a molecule because a single mistake in the generative process renders the entire SMILES string invalid.

Since standard RNN cells lack the ability to count, Θ_M^{Dec} relies on a differentiable version of a stack memory [214], a stack-augmented RNN cell as proposed by *Joulin and Mikolov* [215]. In their work, it was demonstrated that standard RNN cells like the LSTM lack the ability to count which becomes increasingly disruptive for longer sequences [215].

In our implementation of Θ_M both the encoder and decoder consist of bidirectional, stack-augmented GRU cells (cf. Figure 4.3). Stack-RNNs complement any RNN cell with a differentiable push-down stack that operates through learnable controllers, op_t at step t with three operations $\{$`PUSH`, `POP`, `NO-OP`$\}$.

$$op_t = \text{Softmax}(W_{op} h_t) \tag{4.7}$$

where h_t is the hidden state, W_{op} is a $3 \times H$ matrix (H being the dimension of hidden state). At each time step the controller probabilities are determined from Equation 4.7 and the stack memory is updated using the learned controller via a multiplicative gating mechanism:

$$\begin{cases} S_t[0] &= op_t[\texttt{PUSH}]\ \text{Softmax}(W_{so} h_t) + op_t[\texttt{POP}] S_{t-1}[1] + op_t[\texttt{NO-OP}] S_{t-1}[0] \\ h_t &= \text{Softmax}(W_i X_t + W_R h_{t-1} + W_{si} S_{t-1}) \end{cases} \tag{4.8}$$

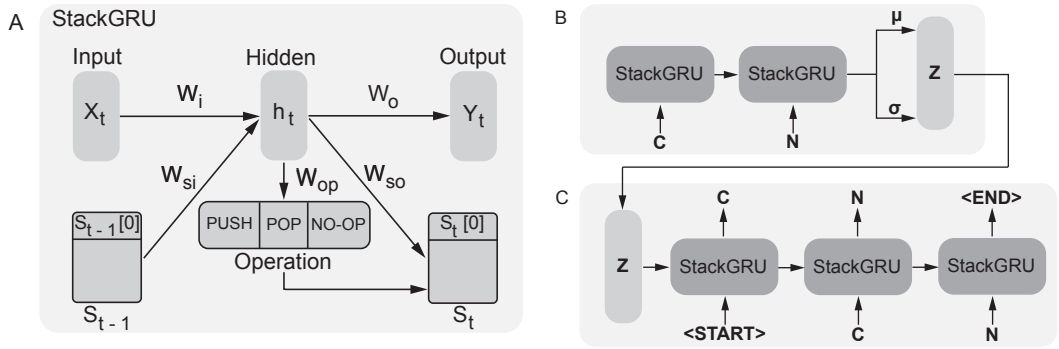


Figure 4.3: Stack-GRU architecture employed in the molecular VAE. A) The StackGRU architecture complements a regular GRU with a differentiable stack that allows one out of three possible operations: `PUSH`, `POP` and `NO-OP`. The operation vector is computed with a softmax from the time point's hidden state. **B)** and **C)** visualize the encoder Θ_M^{Enc} and decoder Θ_M^{Dec} of the molecular VAE respectively. **B)** encodes the SMILES sequences into multivariate Gaussians with parameters μ and σ. **C)** The decoder StackGRU units reconstruct the SMILES sequence from a latent representation ($\mathbf{z_c}$) sampled from the multivariate Gaussian.

where S_t is the stack, W_{so} is a $1 \times H$ matrix and W_{si} is a $H \times N$ matrix (N being the stack height). W_i is the input matrix applied to the sequence and W_R is the recurrent matrix. For brevity this only shows the update equation for the topmost element of the stack.

4.3 *De novo* molecular generation against cancer cell lines

In this section we will describe an application of PaccMann^{RL} to anticancer drug discovery. The goal is to develop a conditional molecular generative model that can be conditioned on a gene expression profile from a malignant tumour and produce a molecule that exhibits high predicted efficacy against that tumour cell profile.

4.3.1 Anticancer drug discovery

Human cancers are subject to intratumoral heterogeneity. They are composed of a collection of single cells with distinct molecular and phenotypic characteristics, leading to highly heterogeneous drug responses in clinical studies [216]. This intricacy is a main factor for the limited number of marketed, targeted anticancer drugs which are usually approved for specific cancer types only. In the past few decades, *de novo* drug design in cancer medicine has struggled to deliver significant advances, partly due to the lack of a holistic approach. Problematically, anticancer drugs are the most challenging therapeutic group – the success rate in clinical trials is at staggering 3.4% [33]. This questions the current methodology for protein target identification. In a seminal study *Lin* et al. [217] investigated ten drug-indication-pairs from ongoing clinical trials and reported that *none* of the ten candidates operated in their proposed MOA. Upon knocking out the ostensible target genes with CRISPR, the anticancer fitness of none of the drugs was impaired, suggesting that they retained their anticancer effect through target-independent mechanisms. *Lin* et al. [217] conclude that off-target toxicity is a frequent MOA of anticancer drugs in clinical drugs which implies that we understand less about the mechanisms of drugs than we think we understand.

Therefore, in this section we propose to use the previously defined PaccMann^{RL} model to generate anticancer candidates solely based on a tumor's metabolic signature (as opposed to attempting to target a specific protein or incorporating information about potential targets directly into the design process). For that study, we will rely on transcriptome data, in particular bulk RNA-Seq gene expression profiles. Transcriptome data has been proven essential to guide lead optimization [218] and has been advocated a pivotal role for *de novo* design [219]. Besides lack of efficacy, off-target cytotoxicity is the main reasons for the high attrition rate in drug discovery [220], suggesting that systems biology might be bridged more closely with drug discovery. Moreover, transcriptome data bears the advantage that it is straightforward to collected whereas identifying protein targets is challenging and time-consuming.

4.3.1.1 Related work

Previous work on using omic profiles to drive molecular generative models is scarce to absent. In a related work, *Méndez-Lucio* et al. [221] proposed a GAN which can be conditioned with a gene expression signature and generate molecules that are likely to *induce* that signature. Their model was demonstrated to generate molecules more similar to existing, active compounds than molecules identified by naive similarity comparison of the expression signatures. Later, *Shayakhmetov* et al. [222] developed a bidirectional adversarial autoencoder that learned a multimodal distribution of molecules and their induced changes in a gene expression signatures. In another related work, *Joo* et al. [223] used a VAE that can be conditioned on a binary IC50 vector defining desired efficacy against cell lines from the NCI-60 database [224] and showed that they could generate fingerprints close to existing anticancer drugs.

4.3.2 Contribution

In contrast to the above work, we here present a RL method for *de novo* molecular design directly from gene expression profiles. Our model can be conditioned on a gene expression signature and generate molecules that exhibit high cytotoxic efficacy (measured as IC50) against that cell profile. Our method incorporates the disease context, a transcriptomic profile, directly into the generative process, and then optimizes the generation toward molecules with high efficacy against a gene expression signature characteristic for a cancer site, a patient subgroup or even an individual, thus constituting a precision medicine methodology. During RL optimization, we employ, PaccMann, a previously published multimodal anticancer drug sensitivity prediction model as reward function [27]. The reward depends on the predicted IC50 between the generated candidate compound and the desired gene expression signature. Without incorporating any specific information about anticancer drugs, the reward function is shown to bias the molecule generation towards molecules with high predicted efficacy against desired gene expression profiles. PaccMann^{RL} is more generic than the aforementioned related work because the model directly generates new molecules and the generative process can be conditioned on any desired gene expression signature. We find that the generated molecules exhibit similarity to known cancer drugs in pharmacological and physicochemical properties (drug-likeness, synthesizability, solubility) and sometimes even show highest structural similarity to existing drugs with known efficacy against these cancer types. While the focus of our experiments is on generating molecules with high drug sensitivity, we note that the reward function is flexible and be refined with subsidiary constraints such as undesired toxicity.

4.3.3 Implementation

We follow the model definition in Section 4.2. For a conceptual overview see Figure 4.1.

4.3.3.1 Gene expression profile VAE

Data. Starting from the TCGA database [225], we build a training dataset of $11,592$ (standardized) RNA-Seq gene expression signatures from healthy and malignant human tissue. The validation data contained $1,289$ samples from the same database. Since the dataset contains $> 20,000$ genes, we used the network propagation scheme described in *Oskooei* et al. [226], originally applied to GDSC [227] by *Manica*, *Oskooei*, *Born*, et al. [27] to select a subset of $2,128$ less-correlated but cancer-relevant genes.

Model. This model corresponds to Θ_C in Subsection 4.2.1. In this case, our "context" samples are gene expression signatures s drawn from the space of gene expression signatures $\mathcal{S}$. To learn a latent representation for gene expression signatures, we used a denoising VAE with four dense layers of $[1024, 512, 256, 200]$ units, `ReLU` activation function and dropout of $p = 0.2$ in both, the encoder and the decoder. The dimensionality of the latent space was $Z_C = 128$. The model minimized Equation 4.1 (a combination of the reconstruction loss and the KL divergence) with Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1\text{e}{-8}$) and a decreasing learning rate starting at 0.001 [77]. For regularization, we employed denoising by 1) applying a dropout of 0.1 on the input genes and 2) adding noise to gene expression values ($\varepsilon \sim \mathcal{N}(0, 0.1)$). The model was trained with a batch size of 64 for a maximum of 2000 epochs.

4.3.3.2 Molecular VAE

Data. The pretraining data was compiled from the ChEMBL database [228] and consisted of $1,576,904$ molecules represented as SMILES that were split in $90\%/10\%$ ratio between training and validation. Molecules that could not be parsed by `RDKit` were removed, the longest molecule had $1,423$ tokens and no padding was needed since our model supported dynamic sequence lengths via `PyTorch`'s packed sequences.

Model. This model corresponds to Θ_M in Subsection 4.2.1. The goal of pretraining this VAE is to learn the syntax of the SMILES language and learn to generate bioactive drug-like molecules. Encoder as well as decoder of this model consisted of two layers of bidirectional GRU (hidden size of 128, dropout of 0.1). Each layer was complemented with 50 parallel, differentiable stacks each with a maximum depth of 50. The dimensionality of the latent space was $Z_M = Z_C = 128$. We relied on teacher forcing [229], i.e., during training the generation of token t is conditioned on the previous ground truth sample as opposed to the token generated at $t - 1$. Whilst this significantly simplifies learning, it may result in posterior collapse [230] which is combatted by a token dropout rate of 0.1 during teacher forcing. The model was trained for 10 epochs with a batch size of 128. To compromise reconstruction loss and KL divergence, we used KL cost-annealing as described in *Bowman* et al. [230].

4.3.3.3 Reward function

The reward function was denoted by $R := f \circ R_I$ in Subsection 4.2.1. The critic $R_I : \mathcal{M} \times \mathcal{P} \rightarrow \mathbb{R}$ is a multimodal drug sensitivity prediction model ingesting a molecule $m \in \mathcal{M}$ and a gene expression signature $s \in \mathcal{S}$ from a cancer cell line. The model was trained using the procedure we reported in *Manica* et al. [231]. The total reward function R is given by:

$$R(m, s) = f(R_I(m, s)) = \exp\left(-\frac{R_I(m, s)}{\alpha}\right) \tag{4.9}$$

where $R_I(m, s)$ returns a log micromolar IC50 value denoting the predicted drug efficacy. Moreover, α is a hyperparameter determining how much the generator is rewarded for proposing molecules with high versus average efficacy, smaller values of α leads to a greedier generator.

Multiproperty optimization. In the case study on multiproperty optimization, we utilized two additional molecular property predictors to compute the reward for the generative model. For this experiment, the reward function was computed as:

$$R_{multi}(m, s) = w_1 \cdot R(m, s) + w_2 \cdot R_{Tox21}(m) + w_3 \cdot R_{SIDER}(m) \tag{4.10}$$

The first part is identical to the reward function above. Let Θ_{Tox21} be the neural network that predicts the toxicity of the 12 Tox21 assays [88] as described in Chapter 2. Then $R_{\text{Tox21}}(m) = 1$ if and only if the output of Θ_{Tox21} is < 0.5 for all 12 Tox21 assays. Otherwise the reward is 0 (as Θ_{Tox21} predicted that m is toxic in at least one assay). Similarly, if Θ_{SIDER} is the network that predicts 27 types of adverse drug reactions [91], then the reward $R_{\text{SIDER}}(m) = 1 - \bar{\mathbf{y}}$, i.e., the inverted mean of the adverse reaction types. Finally, $\vec{\mathbf{w}}$ holds the weights to compute the reward as the weighted sum of the three individual components. We set $w_1 = 1$, $w_2 = 0.2$ and $w_3 = 0.1$. Θ_{Tox21} and Θ_{SIDER} are parameterized using the ToxSmi model described in Chapter 2.

4.3.3.4 RL optimization

Data. To optimize $\mathcal{G}_\Theta$, we used gene expression signatures available from GDSC [227] and CCLE [232] databases. Since these are cell line databases whereas the PVAE was pretrained on human samples from TCGA [225], we validated the standardized gene expression distributions across the databases and found good agreement, confirming the reported consensus between transcriptomic data in CCLE and TCGA [233].

Model. During RL optimization, Equation 4.5 was maximized with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1\text{e}{-4}$, weight decay $1\text{e}{-4}$) and a decreasing learning rate starting at $1\text{e}{-5}$. The gradients were clipped to 2 to prevent $\mathcal{G}_\Theta$ from "forgetting" its chemical

knowledge about SMILES syntax obtained through pretraining on ChEMBL. The reward function hyperparameter α was set to 5. All models were implemented in PyTorch 1.0 [159] and trained on a cluster equipped with `POWER8` processors and a `NVIDIA Tesla P100`.

4.3.4 Results on omic-specific molecular generation

Here, we exemplarily show results for molecular generation optimized against four different types of cancer: lung cancer, prostate cancer, breast cancer and neuroblastoma. For each cancer type, we used 80% of the cell lines (breast: 50, lung: 169, prostate: 7, autonomic ganglia: 56) as training cell lines for the RL optimization. Our baseline molecular generator was the "unbiased" molecular VAE which gives molecules from the chemical space learned from ChEMBL during pretraining. For the evaluation, molecules with a predicted IC50 $< 1\mu$M (i.e. pIC50 > 6) were considered *effective*.

Over the course of RL optimization the generator produced more molecules with high reward, i.e., high predicted anticancer efficacy. To evaluate the generalization capabilities we used 20% of cell lines (per site) for conditioning. The results are presented in Figure 4.4 (left column) and show that our model learned to produce molecules with lower average IC50 values, also for unseen cell lines from a given cancer type. The density plots show that the RL optimization leads the model to generate molecules with a higher mean pIC50 for the target cancer. In each case, a significant portion (between 16% and 57%) of molecules generated from the optimized model had a predicted IC50 $< 1\mu$M, whereas the baseline model only generated 2-5% effective molecules. In the second column of Figure 4.4 we display generated molecules that are predicted to be effective against an unseen cell line from the cancer type of interest. Instead, in the third column we show a precision medicine regime where molecules are shown that were designed a single, characteristic gene expression signature (the mean of all signatures of that side). These molecules were predicted to be effective against the majority of all cell lines from that site as well as against the average profile. Since, according to our knowledge, the formulated problem of conditional molecular generation based on gene expression profiles, has not been tackled before, comparison to previous work is not possible.

4.3.4.1 Investigation of nearest neighbors

In the last column of Figure 4.4 we show one of the top-3 neighbors of the molecules generated in the third column. Similarity was computed using the Tanimoto similarity τ of ECFP4s from several hundreds of existing anticancer compounds [227]. Note that Tanimoto similarity is correlated with induced sensitivity patterns on cancer cell lines [234]. The example breast cancer candidate resembles a collection of fused sugar-like moieties and has doxorubicin, a commonly used chemotherapeutical against breast cancer [235], as one of the top-3 nearest neighbors. Moreover, the lung cancer candidate

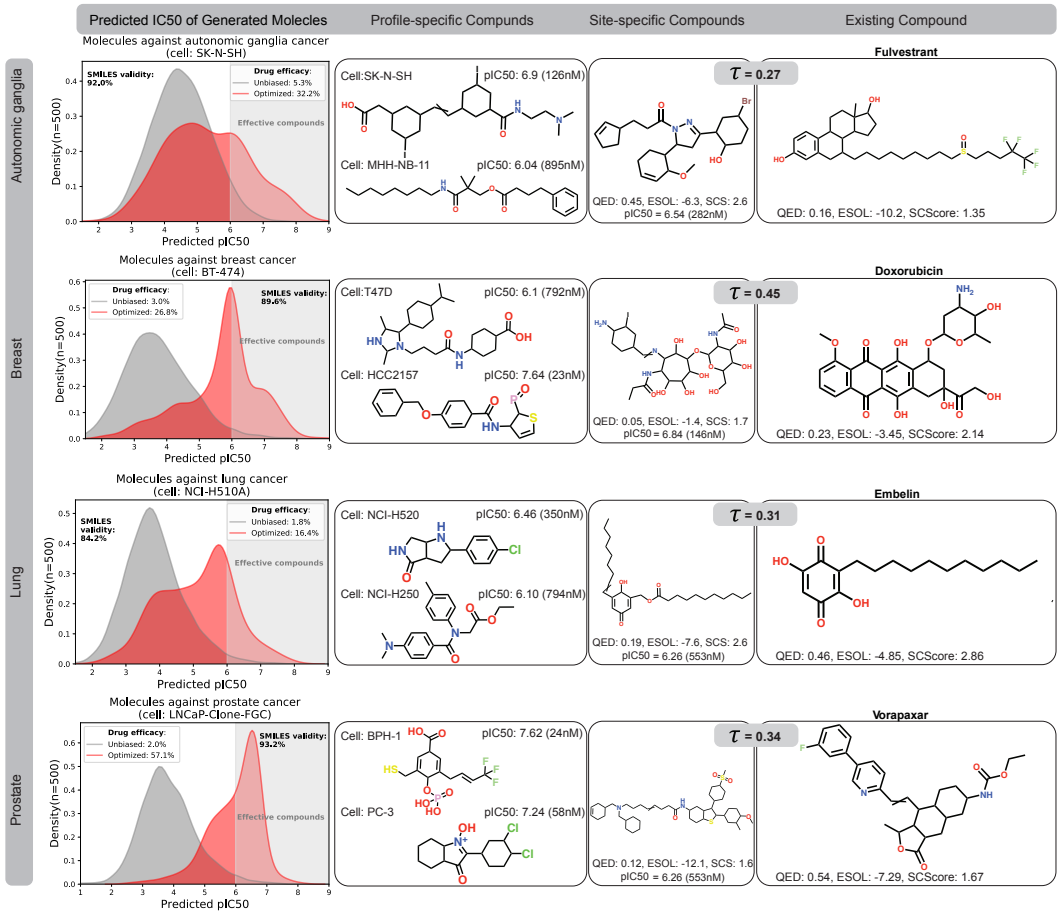


Figure 4.4: Molecular generation results based on unseen gene expression signatures. Each row shows the results of PaccMann^{RL} on a different cancer site. In the first column, the pIC50 distributions of 500 molecules generated with the optimized model are compared to 500 molecules from the unbiased generator. In the second column, we show a generated molecule with low predicted IC50 against a particular cell line, unseen during training. In the third and fourth column we compare molecules that were predicted to be effective against *all* unseen cell lines of a given cancer site with a similar, *existing* anticancer compound.

shows similarities to embelin, an existing anticancer compound from GDSC. Comparing the two structures, it is evident that the generated compound and embelin share a long carbon chain and a single six-membered fully carbonic ring. Embelin was tested against 965 cell lines from GDSC/CCLE from which the highest reported efficacy is against a lung cell line (NT2-D1). Embelin is also known to be the only known non-peptide inhibitor of XIAP [236], a protein that plays an important role in lung cancer development [237]. The closest neighbor of the prostate-cancer molecule in the fourth row is vorapaxar. According to GDSC/CCLE, its efficacy is highest against a prostate cancer cell line (DU_145). Vorapaxar is an antagonist of a protease-activated receptor (PAR-1) that is known to be over-expressed in various types of cancer, including prostate [238]. Lastly, the third most similar compound to the generated neuroblastoma molecule is fulvestrant, an antagonist/modulator of ERα. Fulvestrant has recently been proposed as a novel anticancer agent for neuroblastoma [239]. The predicted pIC50 profiles of our molecule and fulvestrant are highly correlated across all cell lines ($\rho = 0.88$), suggesting that they may exhibit similar pharmacological properties. Similarly, the lung and prostate cancer molecules have also highly correlated activity profiles to their neighbors embelin and vorapaxar ($\rho = 0.55$ and $\rho = 0.69$). In all four cases, the proposed molecules showed the highest structural similarity to existing anticancer compounds that are, either 1) already FDA approved (breast), 2) known inhibitors of relevant targets (lung, prostate) or 3) have been advocated for (neuroblastoma). This positive result is surprising given that the generator was never trained on any anticancer compounds. Only the multimodal drug sensitivity reward function was trained on anticancer compounds.

Broader comparison. In the above analysis the nearest neighbor analysis was restricted to compounds from GDSC/CCLE with known anticancer effects and our analysis focused on verifying that our generated molecules had highest similarity to compounds related to *that cancer type*. In this section, we expand the nearest analysis to a broader chemical space (ChEMBL), aiming to assess whether the most similar compounds are *generally* related to cancer.

With a Tanimoto similarity of $\tau = 0.54$ the nearest neighbor of the breast cancer molecule is CHEMBL1093122, a conjugate of phenyl-2-amino-1-thioglucoside and plumbagin which is known to inhibit the synthesis of mycothiol [240]. Plumbagin as well as many of its derivatives are heavily studied anti breast cancer compounds [241, 242, 243].

For the lung cancer molecule, the nearest neighbor with $\tau = 0.48$ is polyoxyethylene dioleate, a surfactant that has, according to *Girsh* [244] for the treatment of eight cancer types including three lung cancer types (lung adenocarcinoma, metastatic lung cancer and SCLC). Moreover, targeted drug delivery systems use polyoxyethylene dioleat against drug-resistant lung cancer [245].

For our prostate cancer molecule, the nearest neighbour ($\tau = 0.31$) is Clinolamide. This compound is patented as a diagnostic compound for several cancer types, including prostate [246]. For the neuroblastoma compound, the nearest neighbor ($\tau = 0.35$) is NSC-715466 which is included in a NCI-60 release [224]. As we found in that database, it inhibits cell growth by $65\% \pm 15\%$, however with a below-average inhibition for cancer types related to neuroblastoma ($57\% \pm 9\%$). Overall, its performance is roughly on average compared to all $53,217$ compounds, which might have prevented further research. The compounds discussed in this paragraph can be found in Figure 4.5.

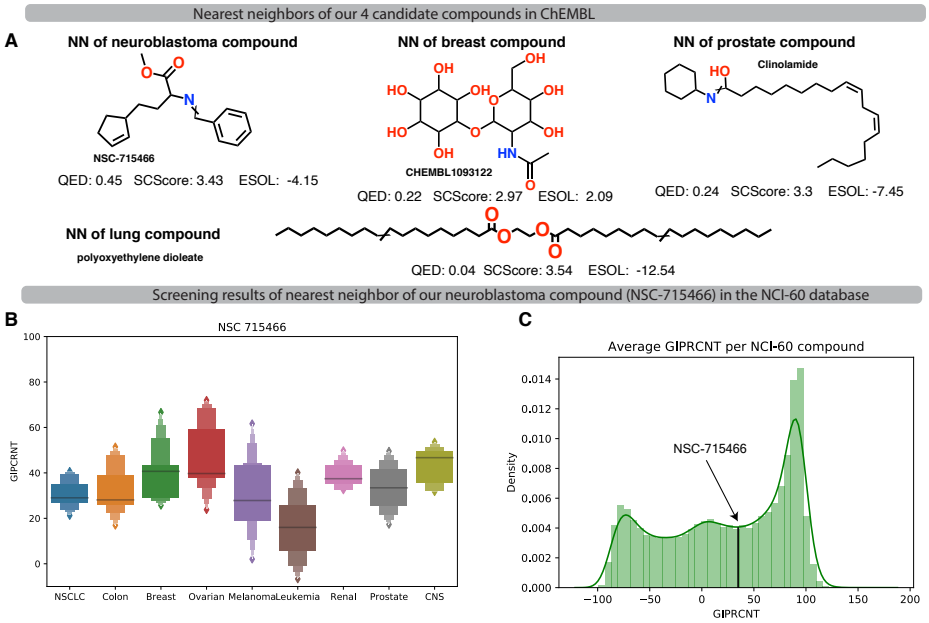


Figure 4.5: Nearest neighbors of our generated molecules (Figure 4.4) in ChEMBL. A) Nearest neighbors and relevant physicochemical properties. **B)** NSC-715466 is part of the NCI-60 [224]. It exhibits, in relative terms, the strongest anticancer effect against leukemia cell lines. GIPRCNT is a cytotoxicity metric (100% means unchanged cell proliferation, 0% complete proliferation inhibition and -100% a full inhibition of all cells. **C)** NSC-715466 showed only moderate anticancer effects.

While the overall evidence in this paragraph is positive (the molecules with the highest Tanimoto similarity to our generated molecules can be linked to cancer, sometimes even to the right subtypes), it has to be emphasized that a high similarity to known cancer drugs does not mean anything *per se*. Oftentimes, even cancer drugs approved for the same cancer subtype or drugs sharing the same MOA exhibit low Tanimoto similarity. For example, across GDSC/CCLE databases, the average Tanimoto similarity ($\tau = 0.149 \pm 0.05$) is not much below the average similarity of two compounds that share the cancer subtype ($\tau = 0.154 \pm 0.06$).

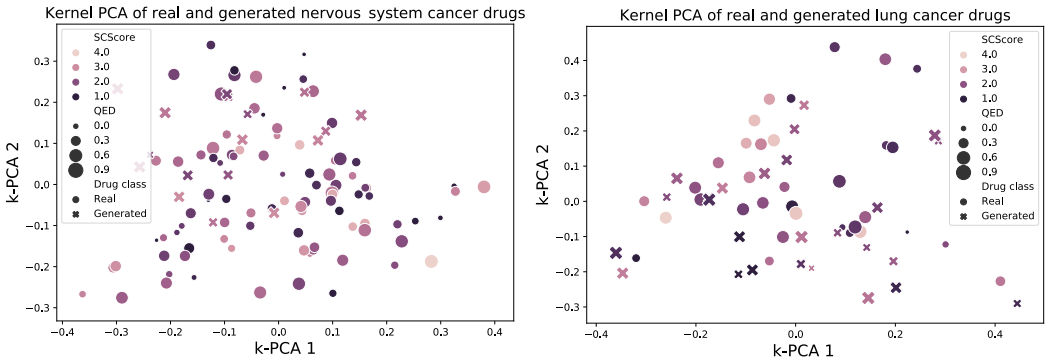


Figure 4.6: Visualization of generated and real molecules. Point size denotes QED score, coloring the SCSCore. Overall, both generated and existing molecules are heterogeneously distributed in the 2D projection and do not form clear clusters.

Aiming to understand whether the generated molecules mimic anticancer drugs, Figure 4.6 shows a kernel PCA [247] of real and generated molecules with Tanimoto similarity as distance metric. We compare both sets of molecules regarding their Tanimoto structural similarity between `RDKit` fingerprints and three relevant chemical properties, namely druglikeness (QED, 0 worst, 1 best), synthetic complexity (SCScore, 1 best, 5 worst) and solubility (ESOL, given in M/L). It can be seen that no clear clusters form, the real and generated molecules are more or less randomly scattered, suggesting that at least some properties of anticancer drugs can be mimicked by our model.

4.3.4.2 Chemical properties of generated molecules

To assess further relevant chemical properties of the generated molecules we compare in Figure 4.7 the distributions of QED [248], ESOL [249] and SCScore [250] between 1) known anticancer compounds (blue), 2) ChEMBL molecules (orange), 3) molecules from the unbiased generator (red) and 4) molecules proposed by the optimized generator (green). Even though these properties were not optimized, comparing the distributions reveals decent agreement overall. Interestingly, compared to the ChEMBL molecules anticancer drugs show lower synthetic complexity scores which might be due to the high attrition rate and resulting policies for cost reduction. Moreover, the unbiased generator generates molecules with more desired properties compared to the training data (ChEMBL). Moreover, the cancer drugs exhibit a significantly lower QED than the other three sets, questioning the usability of this metric. Regarding SCScore, both generators produce molecules with higher complexity than the anticancer drugs (MWU, $p < 0.01$), even though they at least produce molecules with lower complexity than the ChEMBL molecules (MWU, $p < 0.01$). In general, the molecules from the optimized generator have less desired properties compared to the unbiased generator. This is not surprising given that the unbiased generator was optimized to mimic

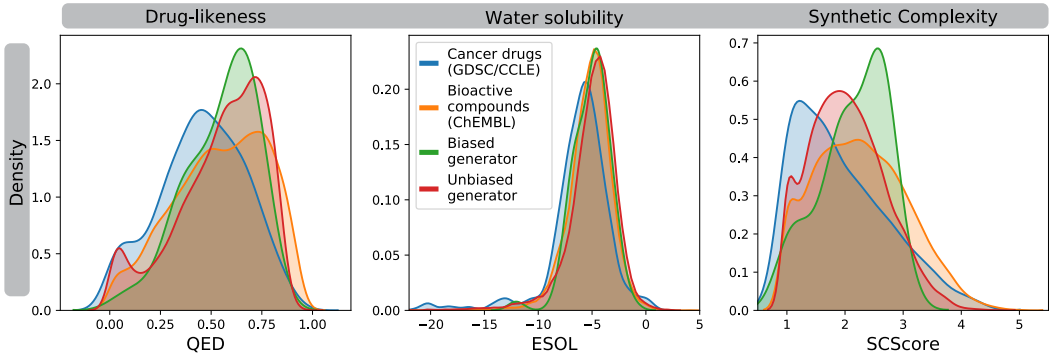


Figure 4.7: Comparison of chemical properties across sets of molecules. We compared three chemical scores for druglikeness as assessed by QED score (0 worst, 1 best), for solubility as assessed via ESOL, given in $\log(M/L)$ (most drugs have a solubility between -8 and -2) and for synthetic accessibility as assessed by SAS (1 best, 10 worst). These three scores are computed for the panel of known anticancer drugs, bioactive molecules from ChEMBL and molecules generated before (red) and after (green) RL optimization.

ChEMBL molecules whereas no explicit optimization was performed during the optimization. A critical property in drug discovery is water solubility; *Savjani* et al. [251] found that 40% of drug candidate have problems with insolubility. While it remains challenging to compute [252] we find an overall high agreement in the ESOL scores of our molecules to generated ones.

4.3.4.3 Validation

Since the model employed as a reward function was trained on anticancer drugs, we sought to verify its generalization capabilities of through a comparison of the predicted IC50 of cancer drugs (most of them were seen during training) and a "negative" set of molecules from ChEMBL across all 965 GDSC cell lines from GDSC. While 15.2% of the anticancer drug "screenings" were promising (IC50$<$ 1μmol), only 7.7% of the ChEMBL screenings revealed such high efficacy. Furthermore, the generated molecules had a higher Tanimoto similarity to existing anticancer drugs than to the ChEMBL molecules ($p < 0.01$, one-sided MWU) or the baseline molecules ($p < 0.01$, one-sided MWU). These encouraging results suggest that the drug sensitivity prediction model can drive the molecular generator in a meaningful direction.

4.3.5 Toxicity: Case study on multi-objective optimization

Besides binding to the protein target or showing high inhibitory effect, novel pharmaceuticals have to fulfill a multitude of other properties. Since the molecular generation in the previous section was solely optimized using a drug sensitivity prediction model, we

assess in this section the quality of molecules generated in a multi-objective optimization setting. Relying on the reward function in Equation 4.10, that additionally includes low toxicity and low adverse drug reaction scores (predicted with ToxSmi, presented in Chapter 2; the results of these models can be found in Subsection 2.7.2), we repeated the RL optimization process.

Figure 4.8 displays a group of molecules generated against the unseen lung cancer cell line NCI-H520 during an RL optimization that was focused on lung cancer. All shown molecules fulfilled the multi-objective, i.e., they had high inhibitory effect against `NCI-H520` (IC50$<$ 1μM) and were not predicted toxic in the Tox21 assays. Moreover, molecules **1** and **2** in Figure 4.8 have a Tanimoto similarity $>$ 0.45 to several FDA-approved lung cancer drugs (e.g., irinotecan, alectinib, vinorelbine, vinblastine, vincristine and topotecan [253]). We also found that molecule **1** showed the highest Tanimoto similarity with vinorelbine, a targeted drug for NSCLC, a type of lung cancer
.

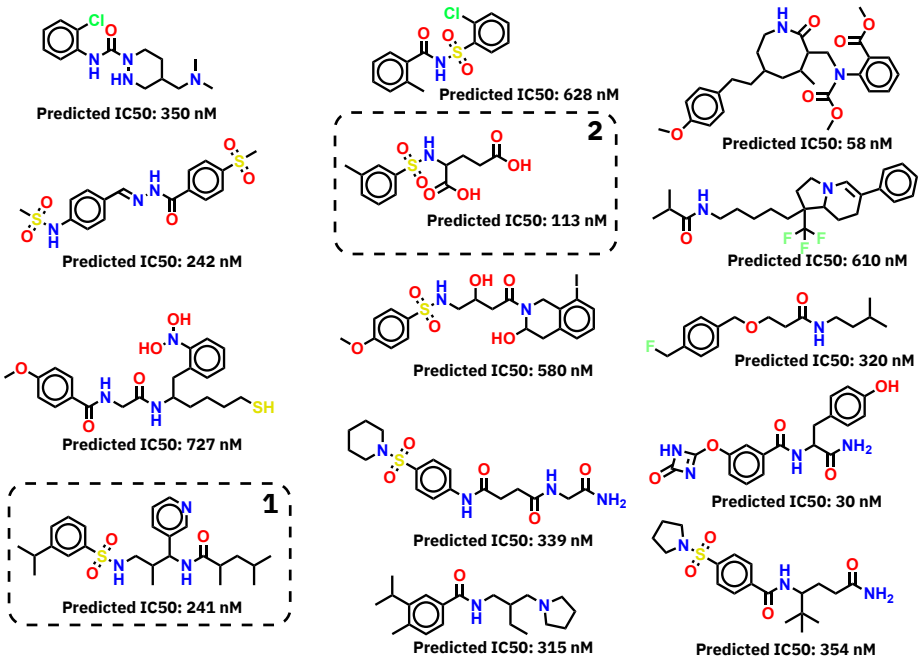


Figure 4.8: Molecules generated in a multi-objective optimization setting. The multi-objective reward was a combination of low IC50, low environmental toxicity and low adverse drug effects. For all shown molecules depicted, all twelve Tox21 predictions are negative.

4.3.6 Discussion

In this section we presented a conditional molecular generative model that can be conditioned on transcriptomic profiles of cancer cell lines. The proposed generative model,

PaccMann^{RL}, was able to produce molecules with high predicted inhibitory effect, even against gene expression signatures unseen during training. Since the molecular generator is driven by a drug sensitivity prediction model, the generator is limited by the predictive power of that model and depends on its generalization capabilities in the chemical space. Note that the molecular generator lacked any cancer-specific domain knowledge as it was only pretrained on bioactive compounds from ChEMBL. We analysed the molecules generated for four different cancer types and found that they shared many structural and functional similarities with known anticancer compounds for the same cancer types that the generation was optimized for. Moreover, we also examined a multiproperty optimization task that included low toxicity as objective and found many candidates with desired predicted properties.

4.3.6.1 Future work

While our methodology constitutes a modest step toward disease-specific molecular generation, future work should include subsidiary properties of drugs into the optimization process. Such multimodal objectives are notoriously challenging to optimize which could be addressed with gradient surgery [254] or maybe even gradient-free global objectives [255]. Future work could also attempt to improve the predictive accuracy of the critic, especially by reducing the distribution shift between the cell lines used for training and the target domain (human samples); a potential solution could be to employ transfer learning as in *Sharifi-Noghabi* et al. [256]. Moreover, for more specific applications than the one presented in this section, a better chemical space could be obtained by finetuning the unbiased molecular generator on more targeted datasets such bioactive molecules with a shared MOA – SMILES augmentation could be used to further improve the latent space quality, in particular if data is limited [67].

4.4 *De novo* molecular generation against SARS-CoV-2 protein targets

4.4.1 Scope

In this section, we will present a case study on accelerated molecular discovery. As visualized in Figure 4.9, we will apply the previously developed method, PaccMann^{RL}, to protein target-driven *de novo* molecular generation. We will focus on 41 proteins related to SARS-

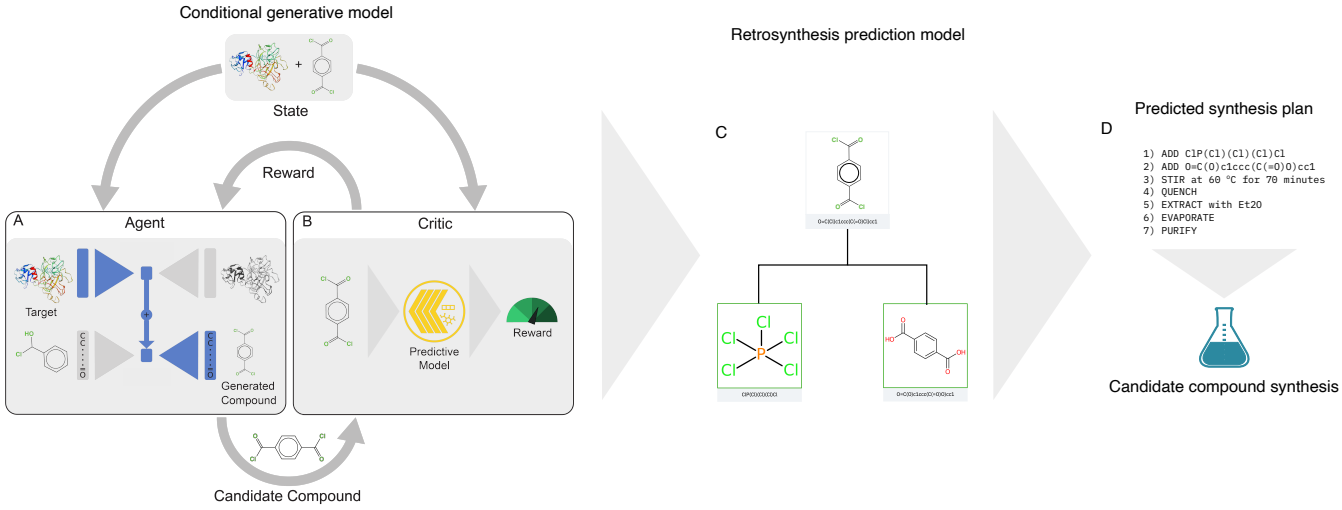


Figure 4.9: A framework for protein target-driven molecular generation and automatic synthesis plan generation. **A**) A molecular generative model that can be conditioned on a target protein of interest. **B**) The generated molecules are evaluated, together with the protein of interest, by a "critic" – a multimodal protein-ligand binding affinity prediction model, additionally coupled with a toxicity penalization (not shown). The molecular generator is trained through reinforcement learning to maximize the the multimodal reward. **C**) Molecular retrosynthesis models are used to find possible synthesis routes for the most promising, generated molecules. **D**) Finally, the predicted synthesis route is converted to a stepwise synthesis protocol that can be executed on an autonomous robotic platform.

CoV-2 and demonstrate with a leave-one-out-cross validation (LooCV) that our method does not require finetuning for specific targets but that it can generalize to proposing ligands with high predicted binding affinity values against unseen targets. The training of the generative model will be guided with a multimodal ligand–protein affinity prediction model (the BiMCA as proposed in Chapter 3) and a toxicity predictor (the ToxSmi model as proposed in Chapter 2). Next, the most promising generated molecules will be evaluated for chemical synthesizability using existing retrosynthesis models. Last, we will

use an automatically generated stepwise synthesis protocol to synthesize a novel, potential inhibitor of the ACE2 protein on a robotic synthesis platform (IBM RoboRXN).

In sum, this captures all aspects from "design" to "make" and shows the feasibility of swift chemical synthesis relying on a completely autonomous workflow.

4.4.2 The search for SARS-CoV-2 antivirals

The emergence of SARS-CoV-2 and the induced COVID-19 pandemic have resulted in long-lasting global health emergencies. Despite the well-understood potential of coronaviruses to cause pandemics [257] as well as previous, related endemics (SARS and MERS), no antiviral drugs against coronaviruses were FDA-approved before the outbreak of the COVID-19. Initial hopes on repurposing drugs such as hydroxychloroquine did not turn out to be effective [258], while others like remdesivir were granted emergency approval [259] and later proved effective at least for oxygen-supplemented patients [260]. However, since the overall success of repurposing strategies has not been satisfying, *de novo* design approaches can be worth an exploration. The common practice in *de novo* drug discovery is to identify a protein target, conceive a potential MOA and then design a ligand to target the identified protein (e.g. an enzyme or receptor). The binding of the ligand to the target can have various effects, e.g., initiating a signaling cascade that ends in apoptosis or docking to a virus' receptor protein that mediates fusion and entry with the host cell such as the spike glycoprotein in SARS-CoV-2.

4.4.3 Related work

Some recent studies compiled virtual libraries of ligands potentially targeting the 3C-like protease, the main protease in SARS-CoV-2 [261, 262, 263]. However, in these works, optimization was performed solely to address the 3C-like protease.

Broadly speaking, the problem of protein-target driven molecular generation was first tackled by *Aumentado-Armstrong* [264], but has received growing attention in the last years [265, 266, 267, 268, 269]. The goal of these works is to generate molecules that can bind to a given protein target (site). From those works, three generated ligands from 3D protein shape [266, 269, 270]. *Skalic* et al. [266] relied on a voxelization of the protein pocket and then employs adversarial training to propose the best-fitting ligand shapes. Instead, *Ragoza* et al. [270] embedded protein and ligand into a joint space, however both works relied on the availability of positive protein-ligand interaction pairs for training. Our approach is related to the sequence-based method proposed by CogMol [265]. Both methods train an independent protein-ligand affinity predictor first, then conditionally sample from a pretrained SMILES generator and can thus, unlike many others, also be applied to *unseen* protein targets. However, *Chenthamarakshan* et al. [265] rely on a conditional rejection sampling method (CLaSS) whereas we use a hybrid-VAE that fuses

protein and chemical latent spaces. The molecules generated by CogMol against three SARS-CoV-2 targets were found to be selective and target-specific in docking studies.

4.4.4 Implementation

To build our conditional generative model, we follow the PaccMann^{RL} model definition in Section 4.2.

4.4.4.1 Protein VAE

Data. To learn the space of proteins $\mathcal{P}$, we retrieved $404,552$ proteins from UniProt [271], discarded large sequences ($>$ 8190 residues) and then embedded all sequences into a 768-dimensional vectorial representation using a 110M parameter BERT-based protein language model that was trained large-scale (on 32.6M Pfam sequences [272] via self-supervised masked-language modeling). This was done in order to not have to train a large-scale protein sequence model.

Model. This model corresponds to Θ_C in Subsection 4.2.1. To learn a latent representation for proteins, we used a VAE with 3 fully-connected layers ($[768, 512, 256]$ units) in both encoder and decoder. Hence the latent space dimensionality $Z_C = 256$. We used ReLU activation function, batch normalization, KL annealing, a dropout of 20%, a learning rate of $3\text{e}{-3}$ and optimized Equation 4.1 with a MSE loss as reconstruction.

4.4.4.2 Molecular VAE

Data. To learn the molecular space $\mathcal{M}$ we use $1,576,904$ molecules from ChEMBL [228], represented molecules as SELFIES [65] that were split in 90% to 10% ratio between training and validation.

Model. This model corresponds to Θ_M in Subsection 4.2.1. The model and training procedure was, unless mentioned here, identical to the one described in Subsubsection 4.3.3.2. Instead of learned embeddings on SMILES sequences, we used used one-hot encodings of SELFIES sequences. The latent space dimensionality was $Z_M = 256$. We used a learning rate of $5\text{e}{-4}$ and 20% token dropout.

4.4.4.3 Reward function

Remember that the reward function was denoted by $R : f \circ R_I$ in Subsection 4.2.1. We set f to the identity function. Let, $p \in \mathcal{P}$ denote a protein from the protein space while, as usual, $m \in \mathcal{M}$ is a molecule from the molecular space. The critic $R_I : \mathcal{M} \times \mathcal{P} \to \mathbb{R}$ is composed by a protein-ligand binding affinity prediction model as proposed in Chapter 3

and a toxicity prediction model as proposed in Chapter 2. Its reward function is given by:

$$R_I(m, p) = R_{Aff}(m, p) + \gamma \cdot R_{Tox}(m) \tag{4.11}$$

where $R_{Aff}(m, p)$ is simply the binding affinity value predicted by $\Theta_{Aff} : \mathcal{M} \times \mathcal{P} \rightarrow [0, 1]$. Θ_{Aff} is detailed in the next section. Moreover, let Θ_{Tox21} be the neural network that predicts the toxicity of the 12 Tox21 assays [88] as described in Chapter 2. Then $R_{\text{Tox21}}(m) = 1$ iff the output of Θ_{Tox21} is < 0.5 for all 12 Tox21 assays. Otherwise the reward is 0 (as Θ_{Tox21} predicted that m is toxic in at least one assay). Θ_{Tox} is parameterized using ToxSmi and trained with SMILES augmentation as described in Chapter 2. Last, let $\gamma \in \mathbb{R}^+$ be a hyperparameter to control the importance of toxicity (we used $\gamma = 0.5$).

4.4.4.4 Binding affinity prediction

Data. The data processing differed from the experiments described in Chapter 3, therefore it is detailed here. We retrieved $1,813,527$ protein-ligand pairs from BindingDB [164], each associated to one out of four metrics (IC50, EC50, Kd and Ki), as visualized in Figure 4.10. Samples with proteins larger than 8190 amino acids

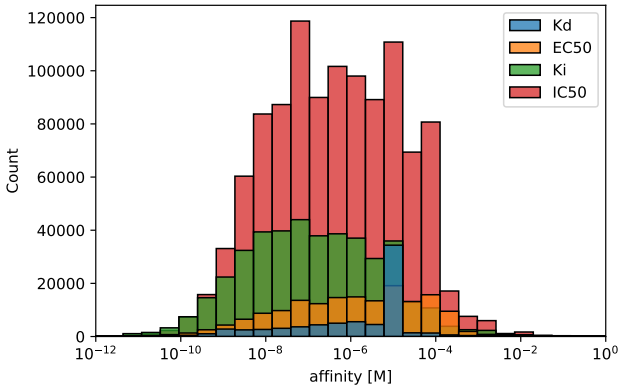


Figure 4.10: Molar affinity values in BindingDB. Molar affinity values as measured by different metrics. 76% of entries report values in the low micromolar range (below 10^{-5}).

were discarded. The remaining $1,361,076$ entries ($7,302$ proteins, $772,634$ ligands) were treated as positive samples. We artificially generated negative samples by randomly assigning 187 compounds to each target which yielded a perfectly balanced dataset of $2,723,726$ samples that was split leniently into chunks of train (72%), validation (18%) and test (10%) data.
We opted for a binary affinity classification (rather than a regression task for three reasons):

1. The hetereogeneity across the four affinity metrics which made it practically impossible to use all samples for training a single regression model,

2. The variety in experimental protocols and conditions used to produce the data gathered in BindingDB that even hampers comparisons of binding affinity values *in the same metric* (see *Kalliokoski* et al. [273] for a detailed report on how this can deteriorate model quality),

3. The large body of previous work framing protein-ligand binding as a classification (e.g., [153, 274]).

As a verification, Figure 4.10 shows the affinity values for all four metrics on a log-molar scale. Since $> 75\%$ of the samples are in the low micromolar range ($< 10^{-5}$) and molecules in the millimolar range are considered uninteresting whereas low micromolar indicates a range that is worth further optimization [275], we decided to treat all samples in BindingDB as positive, especially also given the noise-level in the data [276].

Model. We model $\Theta_{\text{Aff}} : \mathcal{P} \times \mathcal{M} \rightarrow [0, 1]$ with the BiMCA as detailed in Chapter 3. The hyperparameters are given in Table 4.1. We used online SMILES augmentation as well as protein sequence flipping to improve robustness.

Parameter	Value
Protein sequence length T_P	8192
SMILES sequence length T_M	1024
Protein embedding size H_P	32
SMILES embedding size H_M	32
Protein 1D conv. kernel sizes	$[3, 11, 25]$
SMILES 1D conv. kernel sizes	$[3, 5, 11]$
Number of protein kernels	$[32, 32, 32]$
Number of SMILES kernels	$[32, 32, 32]$
Protein attention size A	16
SMILES attention size A	16
Dense layer sizes	$[64]$
Activation function	ReLU
Dropout	30%

Table 4.1: Hyperparameters of the BiMCA.

4.4.4.5 RL optimization

Data. We collected 41 protein targets related to SARS-CoV-2 from UniProt. The full list can be found in the results (Table 4.3). Note that only 9out of the 41 proteins are available in the training data of the affinity predictor Θ_{Aff}, and 27 are available in the training data of the protein VAE Θ_C.

Model. Following Subsection 4.2.1, the conditional generator $G_\Theta = [\Theta_M^{Dec} \circ \Theta_C^{Enc}] : \mathcal{C} \to Z \to \mathcal{M}$. The hyperparameter γ in Equation 4.11 is set to 0.5 (optimizing toxicity is a secondary objective). On the 41 proteins we perform a LooCV for conditional generation, i.e., we train on 40 samples and evaluate on the remaining target. The gradients were clipped to 2 to prevent catastrophic forgetting. Multinomial sampling was applied with a temperature value of 0.8.

4.4.5 Results on targeted molecular generation

4.4.5.1 Validation of binding affinity prediction model

The results of the BiMCA model for protein-ligand affinity prediction on validation as well as test data are displayed in Table 4.2. The BiMCA learned to classify with high accuracy the binding of given protein-ligand pairs. Given that our application is on designing antiviral molecules, we verified that the BiMCA generalizes well to viral proteins. The performance on 10k held-out samples of viral proteins is shown in Table 4.2 (last column) and confirms the good generalization which makes it suitable to be used by the conditional generator during RL optimization.

	Validation	**Test**	**Viral**
ROC-AUC	0.968	0.969	0.96
Average precision	0.963	0.965	0.92

Table 4.2: Result of BiMCA on binary classification of BindingDB affinity samples.

4.4.5.2 Conditional generation

Leave-one-out cross-validation. The main goal of the experiments in this section of the thesis is to verify whether our conditional generative model can go beyond current approaches for protein target-driven molecular design [12, 261, 265]. Specifically, we aim to obtain a model that does not require optimization for a specific target and that, ideally does not even rely on the availability of binding affinity data for the protein of interest. We investigated the generalization abilities with a leave-one-out-cross-validation (LooCV) on the 41 targets. The baseline model is constituted by unconditional sampling of $3,000$ molecules from the pretrained molecular VAE and predicting binding affinities and toxicity scores. For each run, the generative model was optimized for 5 epochs and 500 molecules were sampled per epoch. As can be see in Table 4.3, the RL optimization led to the generation of more compounds with higher predicted binding affinities.

Target protein	Affinity_0	$\text{Aff}_{\text{med}}\pm\text{SEM}$	Aff_{best}	$\text{Tox}_{\text{med}}\pm\text{SED}$	Tox_{best}
VME1-CVHSA	20%	18% ± 3%	**29%**	6% ± 3%	19%
IMA1-HUMAN	88%	97% ± 1%	**100%**	5% ± 3%	18%
VEMP-SARS2	**29%**	16% ± 2%	20%	9% ± 2%	12%
NS7B-SARS2	25%	30% ± 5%	**33%**	7% ± 5%	25%
ITAL-HUMAN	24%	16% ± 6%	**43%**	9% ± 1%	12%
NCAP-CVHSA	**17%**	11% ± 1%	15%	12% ± 2%	14%
R1AB-CVHSA	58%	90% ± 2%	**91%**	9% ± 1%	11%
NS8B-CVHSA	9%	12% ± 2%	**20%**	7% ± 4%	25%
A0A663DJA2-SARS2	26%	35% ± 3%	**41%**	14% ± 3%	18%
NS8A-CVHSA	21%	47% ± 4%	**55%**	10% ± 1%	10%
NS7A-SARS2	4%	3% ± 1%	**7%**	10% ± 3%	19%
Y14-SARS2	17%	29% ± 4%	**43%**	8% ± 2%	14%
NS6-SARS2	20%	12% ± 3%	**22%**	4% ± 3%	14%
SMAD3-HUMAN	50%	74% ± 3%	**86%**	6% ± 1%	10%
SPIKE-CVHSA	3%	0% ± 1%	**5%**	7% ± 1%	11%
DDX1-HUMAN	9%	14% ± 2%	**20%**	9% ± 1%	10%
AP3A-SARS2	**4%**	0% ± 1%	3%	9% ± 3%	19%
R1A-CVHSA	14%	45% ± 3%	**50%**	9% ± 1%	11%
NS8-SARS2	7%	10% ± 3%	**18%**	10% ± 1%	15%
PHB2-HUMAN	4%	3% ± 0%	**4%**	11% ± 3%	23%
SGTA-HUMAN	11%	12% ± 1%	**13%**	8% ± 1%	12%
NS7A-CVHSA	18%	35% ± 5%	**59%**	11% ± 2%	15%
ORF9B-CVHSA	9%	11% ± 2%	**17%**	6% ± 1%	11%
R1A-SARS2	62%	82% ± 3%	**89%**	8% ± 2%	14%
Y14-CVHSA	14%	15% ± 2%	**23%**	11% ± 2%	15%
ORF9B-SARS2	**18%**	12% ± 1%	15%	12% ± 2%	16%
TMPS2-HUMAN	6%	5% ± 1%	**6%**	6% ± 1%	10%
BST2-HUMAN	10%	5% ± 3%	**16%**	10% ± 2%	14%
NS3B-CVHSA	25%	23% ± 2%	**29%**	12% ± 1%	15%
SPIKE-SARS2	7%	6% ± 2%	**12%**	10% ± 1%	12%
FURIN-HUMAN	28%	27% ± 4%	**36%**	9% ± 3%	20%
AP3A-CVHSA	**9%**	0% ± 1%	6%	8% ± 1%	12%
VME1-SARS2	15%	16% ± 3%	**27%**	6% ± 2%	14%
NS7B-CVHSA	21%	26% ± 1%	**27%**	7% ± 1%	11%
MPP5-HUMAN	5%	9% ± 2%	**11%**	15% ± 2%	16%
ACE2-HUMAN	51%	77% ± 4%	**85%**	5% ± 2%	12%
VEMP-CVHSA	21%	25% ± 3%	**30%**	12% ± 2%	20%
NS6-CVHSA	10%	13% ± 1%	**15%**	3% ± 3%	14%
PHB-HUMAN	3%	0% ± 1%	**3%**	6% ± 1%	7%
R1AB-SARS2	83%	100% ± 0%	**100%**	5% ± 1%	7%
NCAP-SARS2	**25%**	5% ± 2%	9%	9% ± 4%	24%
Average	**18%**	**26% ± 4%**	**33%**	**9% ± 0.5%**	**15%**

Table 4.3: Results of leave-one-out cross validation on conditional generation against 41 SARS-CoV-2 targets. Affinity_0 displays the baseline (percentage of binding molecules sampled *before* training), Aff_{best} displays the results at the best epoch of RL training and $\text{Aff}_{\text{median}}$ the median across all 5 training epochs. Same legend for the Tox^{X} columns, but note that Tox_0 was 8.7% in all cases. Per row, it is bolded which model performed best. SEM stands for standard error of the mean.

In detail, in 35 out of 41 cases, the optimized generative model proposed more molecules predicted to bind against an *unseen* target than the baseline VAE. On average, the ratio of binding molecules soared from 18% to 26%. Exemplary density distributions for 2 out of the 41 models are displayed in Figure 4.11. While our reward

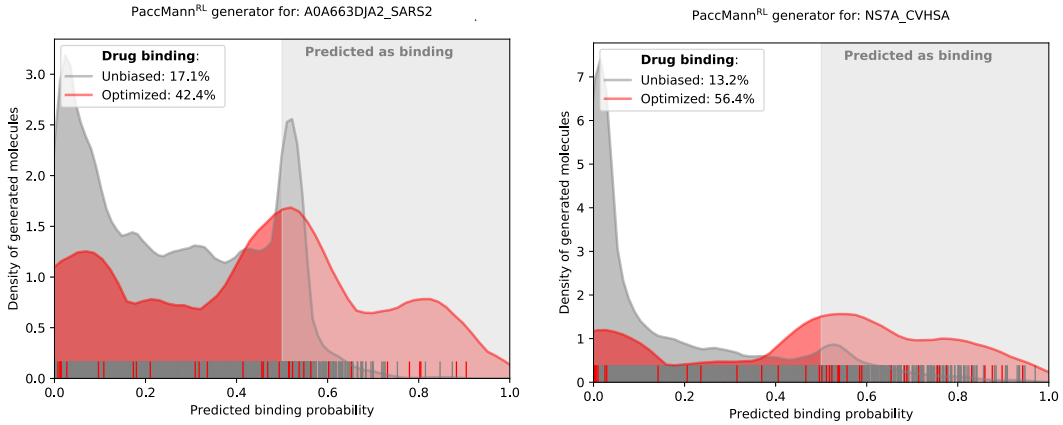


Figure 4.11: Distributions of predicted affinity values during RL optimization. The baseline distributions are shown in gray (affinities of n=3,000 molecules sampled before RL optimization). In red we show the distributions obtained by sampling from the optimized conditional generative model. Clearly, the optimization biased the generative process toward ligands with higher predicted binding affinities.

function (Equation 4.11) also included a toxicity constraint, we note that we did not succeed at reducing toxicity significantly. We hypothesize this was partly because it was a secondary objective and partly because this property is independent of the model input (the protein). To evaluate the molecules qualitatively, Figure 4.12 displays a collection of the generated molecules alongside their QED score [248].

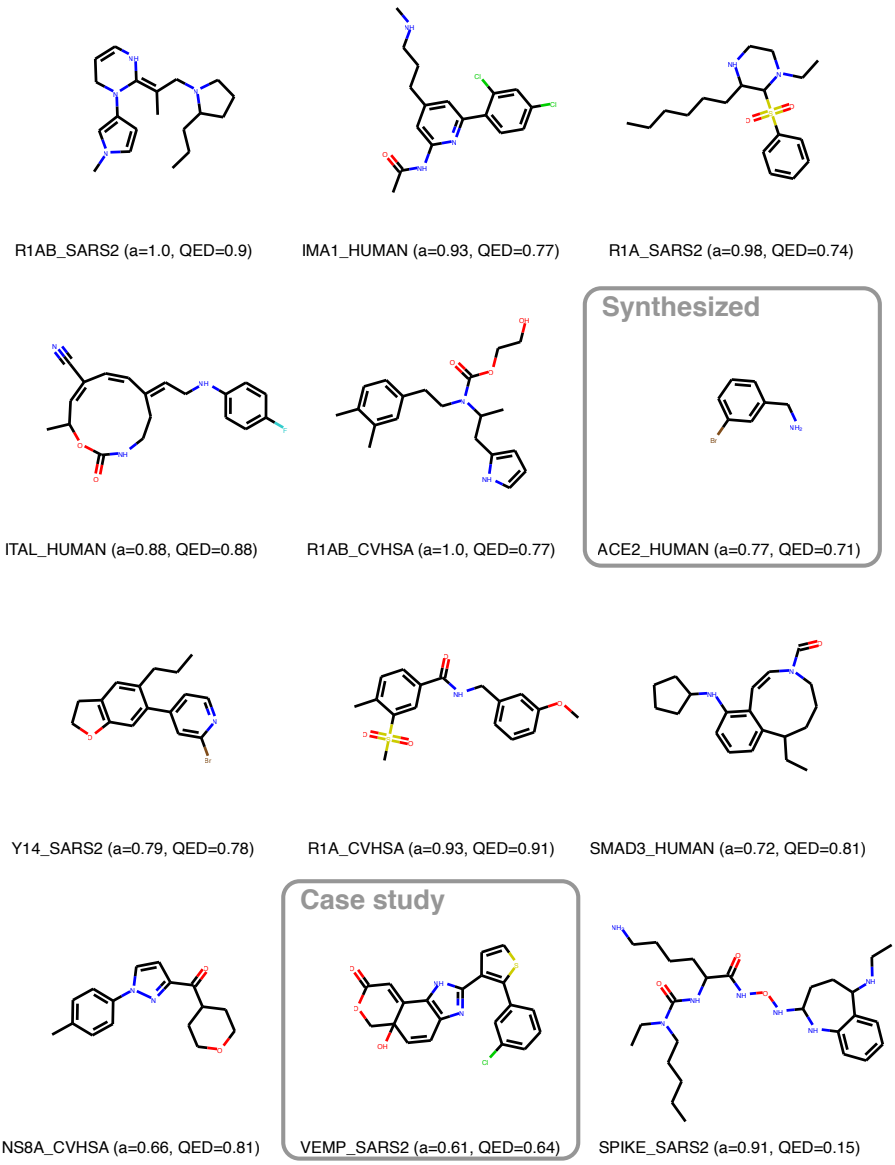


Figure 4.12: Molecules sampled against specific protein targets. For a selection of targets, the generated compound with the highest reward is depicted. a stands for binding affinity. The molecule against `VEMP_SARS2` is further discussed in a case study and the molecule against `ACE2_HUMAN` was synthesized (for details see text)

Learned chemical space. Previous work demonstrated that molecular generative models can reproduce large fractions of the chemical space [277], and thus we aimed to investigate the learned chemical space more systematically. Therefore, we compared four sets of molecules.

1. 10,000 random molecules from ChEMBL
2. 3,000 molecules from our baseline (molecular VAE *before* optimization)
3. 3,000 molecules generated during the optimization
4. 82 SARS-CoV-2 candidate drugs (69 identified by *Gordon* et al. [278] and 13 additional from PubChem [279])

We predicted the binding affinities as well as other physicochemical and pharmacological properties (QED, SCScore, logP and more) and then performed a UMAP [280] on the ECFP4 fingerprints. The Faerun [281] visualization can be found in Figure 4.13. Interest-

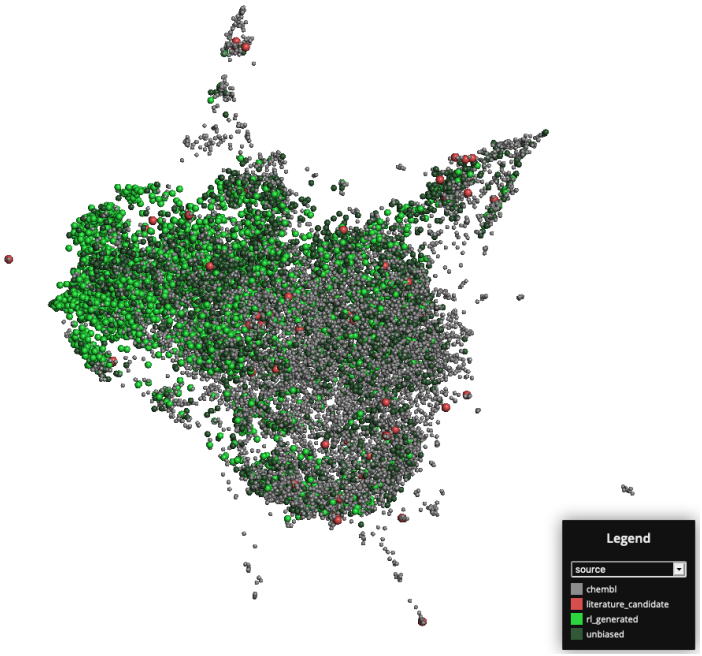


Figure 4.13: UMAP dimensionality reduction of the ECFP4s of different molecule sets. Interactive visualization available at: https://paccmann.github.io/assets/umap_fingerprints.html.

ingly, the optimization steers the compound sampling toward a manifold of the chemical space which is more densely populated with binding compounds. While the literature molecules (red) are structurally fairly dissimilar (no clustering), this plot reveals that the generator succeeded in the objective to generate more molecules with high affinities.

Target selectivity. Selectivity to the desired protein target is a critical aspect in drug discovery because ligands can, on average, be expected to bind to around a dozen different targets [282]. Hence, increasing target selectivity has been identified as a key component to lower the attrition rate [283]. To assess the selectivity, we computed a promiscuity score $P_{m,T}$ for each molecule m and a number of protein targets T by measuring the percentage of targets to which the molecule is predicted to bind. Thus the promiscuity score is an inverted selectivity score. The same set of molecules as described above were used and the results are shown in Figure 4.14. Interestingly, in Figure 4.14 (*left*),

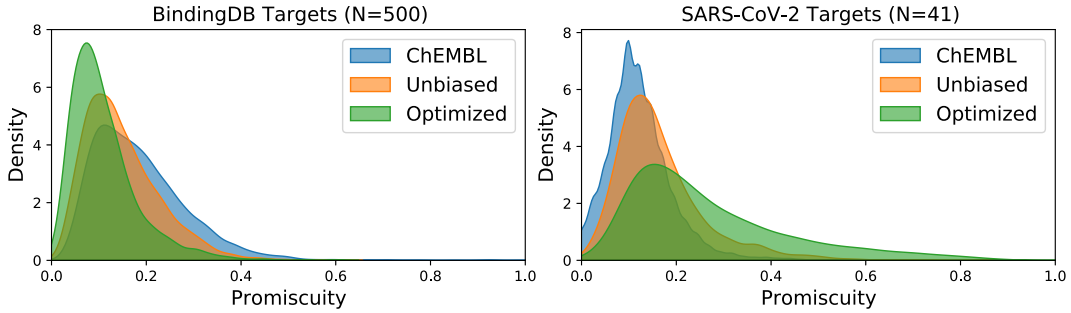


Figure 4.14: Promiscuity of molecules. *Left*: Predicted promiscuity of different molecule sets against 500 random proteins from BindingDB. *Right:* Promiscuity of the same molecules against the 41 SARS-CoV-2 related targets.

where the promiscuity against a random set of BindingDB proteins is shown, the optimized molecules were significantly less promiscuous (i.e., more selective) than both the ChEMBL and the baseline ("unbiased") molecules (Tukey's HSD test, $p < 0.001$ in all pairwise differences, mean for Optimized: 0.11, mean for ChEMBL: 0.19). Keep in mind that no explicit penalty for promiscuity was in place.

Moreover, upon comparing the promiscuity on the 41 examined SARS-CoV-2 proteins, we find that promiscuity is significantly *higher* for the optimized molecules than for the other two sets (Tukey's HSD test, $p < 0.001$ in all pairwise differences, mean for Optimized: 0.27, mean for ChEMBL: 0.12). This finding suggests that our optimized molecules are not only less prone to off-target binding effects but also more likely to bind to related SARS-CoV-2 targets than the other sets of molecules.

Case study. To assess whether our optimized molecules bear some similarity to compounds considered for COVID-19 treatment in the literature, we ranked all 3,000 optimized molecules by their Tanimoto similarity to the closest neighbour of the 82 literature compounds. The top 5 contained the molecule encircled in Figure 4.12. This molecule was generated against $\text{VEMP}_{\text{SARS2}}$ which is the envelope small membrane protein (E-Protein), a key protein for virus assembly and morphogenesis. From all 82 literature candidates, our generated molecule exhibits the highest Tanimoto similarity to the com-

pounds MZ1 and dBET6 ($\tau = 0.64$). Surprisingly, both these compounds were identified by *Gordon* et al. [278] as targeting the E-protein – the protein which was used to condition the generation of our compound.

4.4.6 From hypothesis to synthesis via automation

As aptly summarized by *Chan* et al. [284], two fundamental questions in molecular discovery are: "*What compound to make next? And – how?*
While the focus on this thesis is certainly on the first question, we acknowledge that the second question is similarly critical. Synthesizability has been identified as a key problem in the field, e.g., *Gao* and *Coley* [285] concluded that the utility of deep generative models is "*stymied by ignorance of synthesizability*".

Therefore, in this subsection we will tackle the critical step of turning a *hypothesis*, i.e., a molecule produced by a generative model with promising predicted properties, into a *chemical* produced in the lab. Notably, to achieve that, we will rely on a completely autonomous workflow that includes retrosynthesis models (to find a synthesis route), synthesis protocol models (to find stepwise instructions for the synthesis execution) and robotic hardware (to execute the synthesis protocol). The four steps that enabled a completely autonomous workflow that led to the successful synthesis of one candidate molecule are:

1) Protein-targeted molecular design with conditional generative models

2) Assessing synthesizability of molecules from **1)** via retrosynthesis route prediction

3) Extracting stepwise synthesis protocols from retrosynthesis routes in **2)**

4) Chemical synthesis using robotic hardware that executes the protocol from **3)**

While we have detailed **1)** already above, we will cover **2)**, **3)** and **4)** below.

Note that the tools for **2)**–**4)** are publicly available via the `IBM RXN for Chemistry`[3] platform and can even be used directly via the Python package `rxn4chemistry`[4]. Please also note that none these tools are part of this thesis, here we simply use them to demonstrate an application on accelerated molecular discovery.

4.4.6.1 Retrosynthesis prediction

The top-5 generated molecules for each protein target were considered for chemical synthesis.

[3] `https://rxn.res.ibm.com`
[4] `https://github.com/rxn4chemistry/rxn4chemistry`

PROCEDURE. Relying on a language-based molecular retrosynthesis model proposed by *Schwaller* et al. [60], we predicted potential synthesis routes for all 205 candidates. This retrosynthesis model builds upon two Molecular Transformers [28], one for forward reaction prediction (precursors to product) and one for backward reaction prediction (product to precursors). Both models are combined with a hypergraph exploration strategy that ranks the sets of predicted reaction (produced via beam search) based on a scoring scheme that uses the confidence score of the forward model (round-trip accuracy). For our experiment, we limited the number of synthesis steps to 6, used a forward confidence threshold of 0.6 and set the beam width to 10.

Altogether, the model predicts a synthesis tree composed of commercially available starting materials (the leaves) and intermediate products (the nodes) that are connected with hyperedges (the reactions).

RESULT. Although the generated molecules are not optimized for synthetic accessibility, for $\sim 30\%$ of the top-5 molecules a successful synthesis route with at most 6 steps was predicted (cf. Figure 4.15**A**). Notably, almost 50% of the feasible molecules require only one or two steps for synthesis, suggesting that many of our molecules can be produced within a few steps from commercially available materials. As shown in Figure 4.15**B**, overall, more than 60% of the > 2000 predicted synthesis routes are feasible (note that for each molecule, multiple routes were generated). The correlation analysis in Figure 4.15**C** between chemical and pharmacological properties further indicates that some properties like QED and synthetic feasibility are very correlated among our molecules.

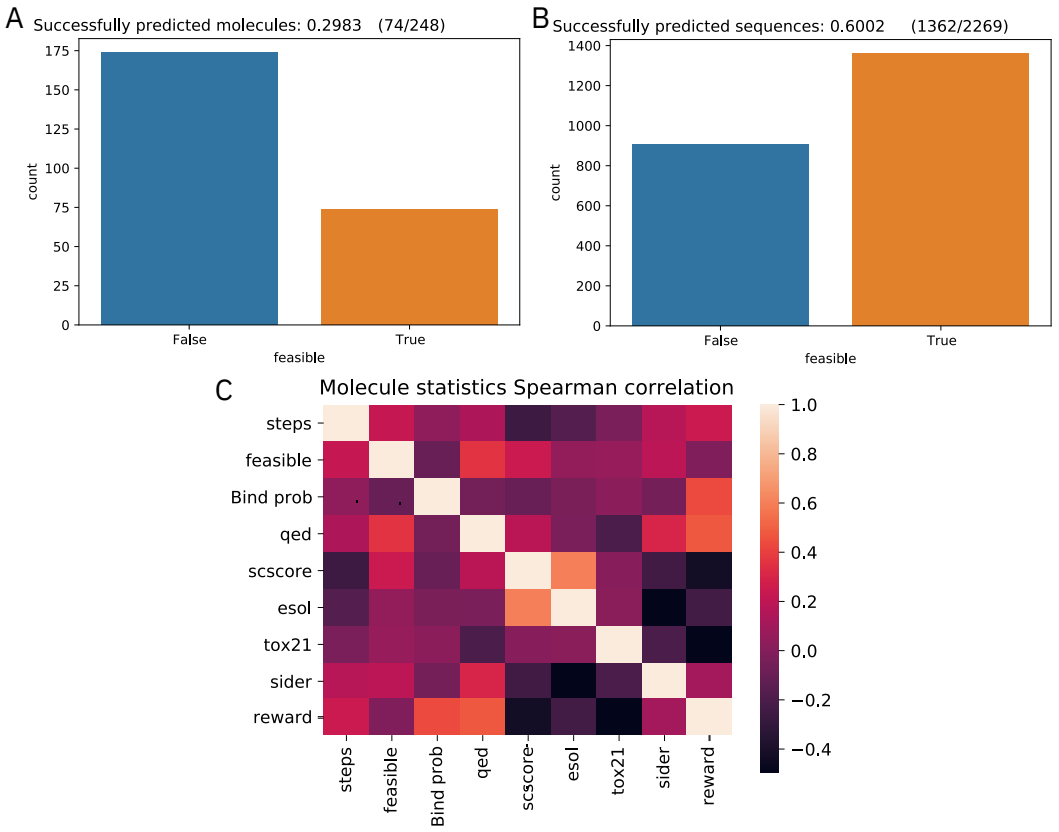


Figure 4.15: Results of retrosynthesis prediction of top-5 molecules per target. A retrosynthetic pathway is considered feasible if, within 6 reaction steps, all precursors are commercially available **A**: For 30% of all generated molecules, feasible routes were found. **B**: Feasibility over all predicted sequences which includes intermediate reactions. **C**: Correlation of synthesis-related properties (e.g., the (binary) feasibility of the synthesis or the number of steps) with several physicochemical properties, e.g., (estimated) water solubility, QED, SCScore etc.

4.4.6.2 Selection of synthesis candidate

Selection of ACE2 target. We decided to synthesize a molecule that was generated against ACE2, a host protein that is widely regarded a promising target for SARS-CoV-2 antivirals [286, 287, 288]. Even though ACE2 was even argued a pivotal role for COVID-19 antiviral design [289], generative models against SARS-CoV-2-related host targets are remarkably absent with the exception of *Ray* et al. [290]. Therefore, we herein aim to fill this gap and exemplify the process of synthesizing a ligand predicted to bind to a host target.

ACE2 is a type 1 membrane protein that is mostly expressed in lung alveolar epithelial and endothelial cells [291, 292, 293]. It is a major player for the regulation of cardiovascular homeostasis [294, 295], the protection from epithelial cell injury [296, 297, 298]. Most

importantly, it has been identified as a functional receptor for SARS-CoV [299, 300] as well as SARS-CoV-2 [301, 302] which mediates cell entry through ACE2 via its spike (S) protein [301]. Due to the importance of the spike protein for viral cell entry, ACE2 inhibitors seem a logical approach and has been highlighted by *Tai* et al. [303] and *Chen* et al. [304].

Selection of molecule. We decided to synthesize 3-Bromobenzylamine to demonstrate an autonomous workflow for discovery and synthesis of potential inhibitors for SARS-CoV-2-related protein targets. We selected 3-Bromobenzylamine due for two main reasons:

1) We performed a maximum common subgraph similarity search [305] of our ACE2-generated molecules to the 82 COVID-19 literature candidates. This analysis identified 3-Bromobenzylamine as a full substructure of Arbidol (Umifenovir) which is a broad-spectrum antiviral drug that has been used in Asia against influenza and hepatitis [306, 307]. Strikingly, Arbidol has been proposed as a COVID-19 antiviral due to its interaction with the ACE2 receptor [308], exactly the target against which 3-Bromobenzylamine was generated by our model. The MOA of Arbidol is that it inhibits the fusion of SARS-CoV-2 with the host cell and thus prevents the virus from entering the cell. This MOA was first conceived [308], and then confirmed in docking studies[309, 310] and *in vitro* experiments [311]. Furthermore, the efficacy of Arbidol in viral diseases is well known [312]. Regarding COVID-19, *Wang* et al. [313] reported that Arbidol decreased the mortality rate and *Wei* et al. [314] found that it reduced the duration of moderate and severe courses of COVID-19 by > 6 days. We therefore believe that 3-Bromobenzylamine, a smaller and more broadly available compound, might operate in a similar MOA against ACE2, especially given that the presence of the bromine in Arbidol was reported as important for the efficacy by *Di Mola* et al. [315].

2) Apart from this relation to Arbidol, the pharmacological properties of 3-Bromobenzylamine are also desirable. The predicted ACE2 affinity was 0.77, the drug-likeness (QED) was high (0.71), like $> 99\%$ of all approved drugs it contains an aromatic ring, the solubility is with -2.66 well in the range of approved drugs (cf. Figure 4.7). The promiscuity (i.e., the probability of off-target binding) to the remaining protein targets is relatively low (0.13) whereas it is relatively high (0.27) for the 40 SARS-CoV-2 related targets (cf. Figure 4.14). 3-Bromobenzylamine passes the Lipinski rule of five [316], has a molecular weight of 186 Dalton and was predicted to be non-toxic in all but one of the twelve Tox21 tasks (NR-AR-LBD, the androgen receptor ligand-binding domain).

Note that 3-Bromobenzylamine is not a *de novo* compound since it has been synthesized before. Obviously, in some occasions, any molecular generative model proposed molecules that already reported in chemical databases.

The predicted retrosynthesis route for 3-Bromobenzlamine is shown in Figure 4.16. Since the predicted synthesis route was quite simple (one-step route) and assigned a high

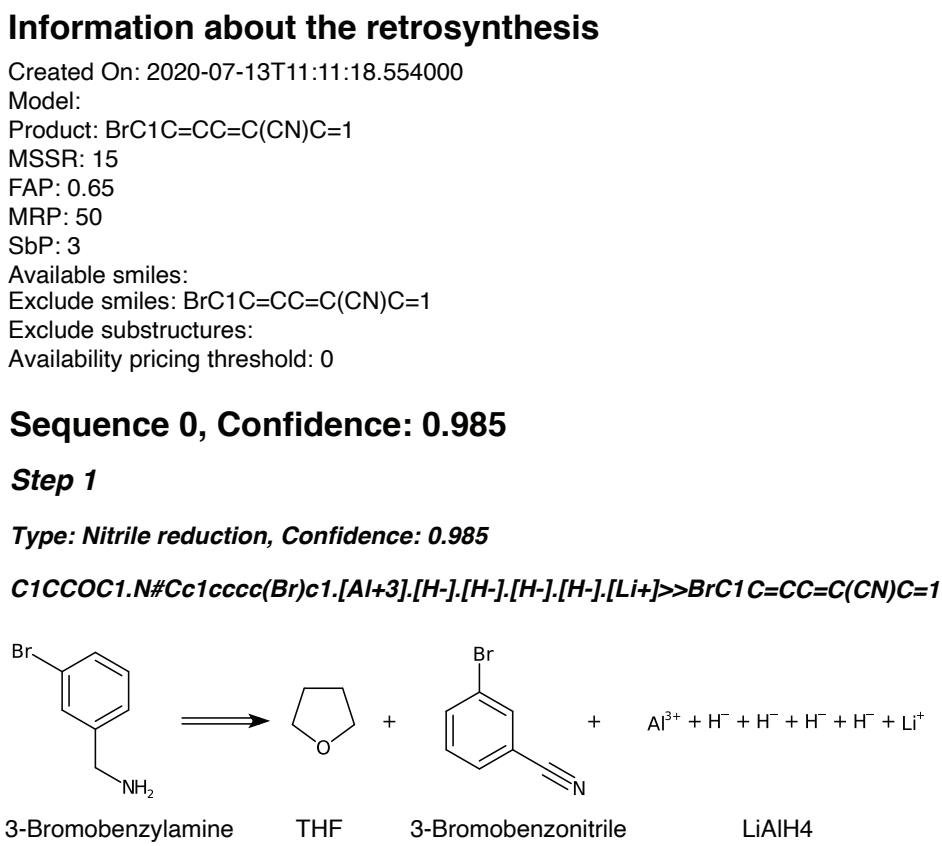


Figure 4.16: Synthetis route for 3-Bromobenzylamine. A predicted one-step synthesis route for 3-Bromobenzylamine. The Nitrile reduction was predicted with very high confidence (98.5%). It reduces 3-bromobenzonitrile using lithium aluminium hydride.

confidence (98.5%), we ultimately decided to synthesize 3-Bromobenzylamine. Even though the nitrile reduction that reduces 3-bromobenzonitrile with lithium aluminium hydride is challenging, it is a known reaction that minimized our risk of complication during synthesis.

4.4.6.3 Synthesis protocol generation

Procedure. Upon selection of the best molecules and their respective routes, we used the method proposed by *Vaucher* et al. [61] to generate a synthesis protocol in natural text.

Their synthesis action generation model was trained on chemical recipes and predicts a sequence of actions from a reaction encoded as SMILES.

Result. The predicted procedure is shown in Figure 4.17. Note that the brine was not

Predicted stepwise synthesis execution protocol

1. **ADD:** `At 25 degree Celsius, add 9 ml of anhydrous THF into a glass reactor of 100 ml.`
2. **MAKE SOLUTION:** `Use 9 ml of a solution of 3-bromo-benzonitrile in THF (0.11 M, 1 mmol).`
3. **ADD:** `Add under gentle stirring (100 rpm).`
4. **ADD:** `While maintaining a temperature of 25 degree Celsius, add 1 ml of LiAlH4 in THF dropwise across 180 seconds.`
5. **STIR:** `Stir for 5 minutes at 25 degrees Celsius.`
6. **QUENCH:** `Quench the excess of LiAlH4 with 2 mL saturated NaCl aqueous solution.`
7. **STIR:** `Stir for 60 seconds.`

Figure 4.17: Predicted synthesis protocol for the synthesis of 3-Bromobenzylamine. For each step in the predicted protocol, we display the action type in bold, followed by the instruction. The protocol was generated using the model from *Vaucher* et al. [61] via the `RXN for Chemistry` platform.

provided directly in the predicted synthetic route, but we favored it over water or an alcohol for quenching since it prevents the formation of a colloidal dispersion of aluminum hydroxide.

4.4.6.4 Chemical synthesis

The reaction was executed on the `IBM RoboRXN` hardware, a platform for completely autonomous synthesis execution. The stepwise synthesis execution protocol in Figure 4.17, generated in the previous section, can be automatically translated to a series of mechanical instructions that can be executed on the robotic hardware. The synthesis was thus executed without the involvement of human labor and after the reaction finished, the organic layer was collected and further analysed. We diluted 0.3 mL of the organic layer 50

times and then analyzed with a liquid-chromatography-mass spectrometry (LC-MS), in particular the Agilent TOF6230. The LC/MS result shows a clear peak indicating the presence of 3-Bromobenzylamine with a score of more than 99%. The screening analysis could not identify signals related to the precursors. The result from the mass spectrometry analysis is reported in Appendix Section A4. Even though this qualitative analysis cannot be used to corroborate quantitative arguments, we believe that the lacking evidence about presence of precursors is an indication that the synthesis of 3-Bromobenzylamine was successful.

4.4.7 Discussion and limitations

In this section we examined our proposed methodology for conditional molecular generation on SARS-CoV-2-related protein targets. As reported in a leave-one-out CV, we demonstrated that it generalize to proposing ligands that are likely to bind against *unseen* protein targets – to the best of our knowledge a feature previously not reported in the literature. Even though we did not optimize selectivity explicitly, the generated molecules showed high selectivity across an unrelated set of proteins. Notably, the same molecules showed the lowest selectivity across the remaining SARS-CoV-2-related targets, suggesting that our model learned some general binding patterns of those targets. An apparent bottleneck in our methodology is the performance of the critic (i.e., the binding affinity predictor) which steers the generative model. Improvements on this model could be made with recently released data (e.g., the $1,670$ compounds screened against SARS-CoV-2 proteins [317]).

Besides a novel, protein-target-driven molecular generative model the main contribution of this section was a demonstration of the feasibility of a completely autonomous workflow – from the *hypothesis* to the *synthesis* all steps were executed without the injection of domain knowledge or manual labor. This was achieved through the integration of molecular generative models with retrosynthesis and synthesis protocol prediction models as well as robotic hardware for automatic synthesis execution. We selected 3-Bromobenzylamine, a substructure of the antiviral drug Arbidol (which has proven effective in treating hospitalized COVID-19 patients [313, 314]) for synthesis on a robotic platform. While this was an interesting proof-of-concept, we emphasize that the true bioactivity of 3-Bromobenzylamine can only be assessed via *in vitro* and *in vivo* experimentation. Moreover, 3-Bromobenzylamine is a commercially available compound and thus not a *de novo* compound *per se*. To validate the potential of the generated molecules, it could have been helpful to perform docking studies with the chosen protein targets prior to synthesis. However, *Di Mola* et al. [315] investigated docking of Arbidol to ACE2 and found that some structures of Arbidol, which are preserved in 3-Bromobenzylamine, are key for binding to ACE2. Moreover, generative models similar to ours have shown that primary structure of SARS-CoV-2-related targets can suffice to obtain molecules with favorable binding free energies [265].

After all, however, the main goal of this section, however, was the demonstration of an autonomous workflow for rapid generation and synthesis of compounds with desired properties *in silico*. One advantage on this aspect is that our method neither hinges upon the availability of screening data for the target of interest, nor on the availability of protein 3D structure. This is especially relevant in the context of emerging diseases [318].

4.5 A short note on multimodal context

In this chapter, we have applied the proposed PaccMann$^{\text{RL}}$ framework to generate molecules that inhibit specific cancer cell lines (represented by their gene expression signature) or bind to specific protein targets (represented by their amino acid sequence).

We emphasize that our model formulation (Subsection 4.2.1) is generic and the *context* used to drive the generation could possibly be multimodal. One may conceive a generic framework where the molecule generation is conditioned on any combination of an omics profiles, a target protein or a molecular scaffold. In the recent work by *Janakarajan, Born,* and *Manica* [319] we proposed a first step in that direction by jointly conditioning on a gene expression signature and a target protein. In that case, the reward function was composed of the predicted binding affinity between the target protein c_p and the proposed molecule m as well as the molecule's predicted pIC50 efficacy against a cancer cell line characterized by the expression signature c_s:

$$R(m, \mathbf{c}) = \text{Affinity}(m, c_p) - \text{pIC50}(m, c_s) \tag{4.12}$$

In that work we proposed a fully differentiable set autoencoder that can embed an arbitrary number of modalities (i.e., items of a set) in a permutation-invariant fashion into a fixed-length vectorial representation that can be used to condition the molecular generator. The permutation invariance in the decoder of the set autoencoder is achieved through a set matching module that approximates the solution to the linear sum assignment problem (i.e., matching the decoded items to the input items). This work is mentioned here for completeness but further detailed in *Janakarajan, Born,* and *Manica* [319].

4.6 Exploring learned chemical spaces via Gaussian Processes

So far, in this chapter, we have seen two applications of PaccMann$^{\text{RL}}$, a molecular generative model that can be controlled with a complex biomolecular context such as target proteins or omics expression signatures. This model relied on a pretrained molecular VAE and then *finetuned* the decoder to generate molecules. This finetuning begs the risk of catastrophic interference, i.e., the decoder might "unlearn" to produce a versatile set of molecules because the reconstruction objective is taken out of the equation. When

searching for molecules with desired properties, an alternative to *finetuning* a generative model is *exploring* its learned chemical space. In this section, this will be assessed with Bayesian optimization (BO) techniques, in particular Gaussian Processes (GP).
Since the chemical space is fixed in this setting, the GP optimization can be seen as a global search in the (learned) chemical space. GPs are interesting for molecular discovery tasks because they facilitate the efficient maximization of functions that are costly to evaluate. Previous work already coupled VAEs with Bayesian optimization for molecular generation [11, 320]. However, while their work focused on the optimization of chemocentric properties like drug-likeness or logP (partition coefficient), we believe that no prior work has optimized a multimodal property like protein-ligand binding affinity.

4.6.1 Methodology

Similar to Section 4.4, the goal in this section is to generate molecules with high predicted protein-ligand binding affinity to a protein of interest. However, unlike in Section 4.4, we will not focus on SARS-CoV-2-related targets but instead, like in Section 3.6, on human kinases. We will rely on the molecular VAE as proposed in Subsubsection 4.3.3.2, i.e., a VAE with a an auto-regressive encoder and decoder trained on $>$ 1M molecules from ChEMBL [321] on the ELBO objective (Equation 4.1). Note that we trained two versions of this model, one on SMILES strings (as detailed in Section 4.3) and one on SELFIES (as detailed in Section 4.4). We will now formulate a GP to approximate the predicted binding affinity of the protein kinase of interest and the generated ligand.

Bayesian Optimization with GPs. Given a protein p of interest as well as our molecular VAE the goal is to find the latent code $\hat{\mathbf{z}}$ that maximizes our affinity prediction $\Theta_{Aff} : \mathcal{P} \times M \rightarrow A$. Remember that, during training, the molecular VAE optimizes the ELBO (Equation 4.1). This means that each sample defines an encoding distribution $q(\mathbf{z}|\mathbf{x})$ which is, thanks to the KL divergence, constrained to be similar to a predefined prior distribution $p(\mathbf{z})$, in our case $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\vec{0}, \mathbf{I})$, i.e., a multivariate unit Gaussian. Considering the sheer size of the latent space and especially the cost to evaluate the function Φ_{Aff} we reformulate the problem as a Bayesian optimization task:

$$\hat{\mathbf{z}} = \underset{\mathbf{z} \in \mathcal{Z}}{\arg\max}[\Theta_{Aff}(p, p(\mathbf{x}|\mathbf{z}))] \tag{4.13}$$

where $p(\mathbf{x}|\mathbf{z})$ is approximated by the decoder. Essentially, the Bayesian optimization performs an iterative search with the objective to minimize the number of calls to Θ_{Aff} such that the stopping criterion is met, i.e., we reached a point $\hat{\mathbf{z}}'$ in the latent space that yields an affinity a such tha $|a - a_{max}| < \varepsilon$. The optimized function is modelled with a prior that specifies a probability distribution over functions, in our case a GP prior:

$$\Theta_{Aff} \sim GP[\hat{m}(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')] \tag{4.14}$$

where $\hat{m}$ is the mean function and k is a kernel denoting a similarity between two points. During optimization it is thus assumed that the affinity prediction function follows a multivariate Gaussian, with negative expected improvement (EI) as acquisition function [322]. This trades off exploration and exploitation to determine the next evaluation point.

Optimization procedure. We selected six kinases for the analysis, namely the four JAK family members (JAK1, JAK2, JAK3, TYK) as well as ABL1 (the target of imatinib, the first FDA approved kinase nhibitor [172]) and MAPK11 (a thoroughly studied target from the MAPK family and isoform of MAPK14/P38α [323]). We performed four experiments per kinase, respectively two on the SMILES and two SELFIES VAE, respectively on using the full protein sequences or only the active site sequences in Θ_{Aff} – which was was implemented through the k-NN model as detailed in Section 3.6. For each kinase, the optimization was initiated from 40 random points in the latent space and performed for 30 epochs with 50 calls per epoch using `scikit-optimize`. After each epoch, we generated 300 molecules from the latent points.

4.6.2 Optimizing binding affinities

The results for the latent space optimization are shown in Figure 4.18. The results on

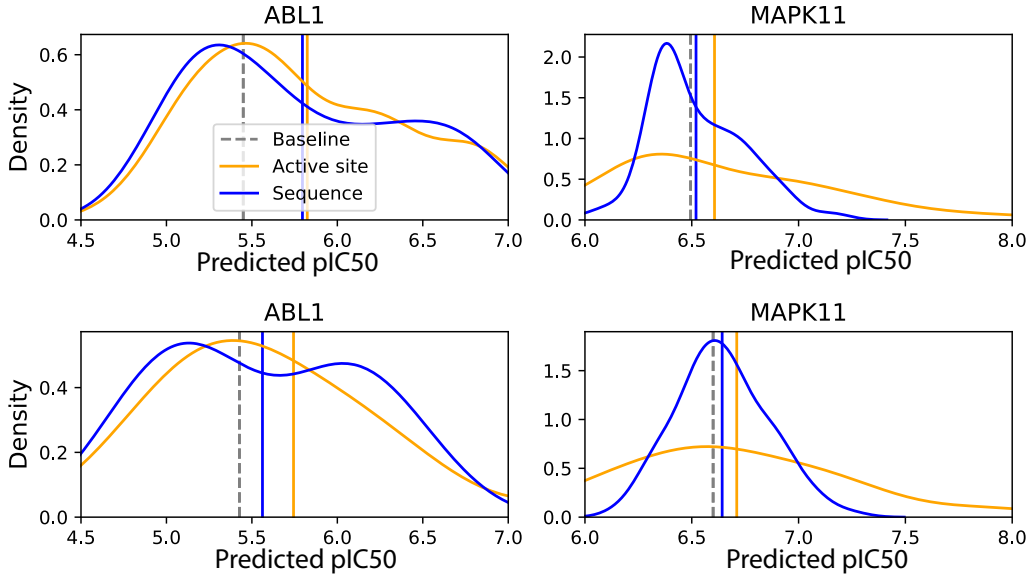


Figure 4.18: GP-based exploration of the latent space to find kinase inhibitors. For the two selected targets, ABL1 and MAPK11, we show the (predicted) pIC50 (higher=better) distribution of generated molecules. The dashed line denotes the mean pIC50 before the optimization. In the top row, the SELFIES generator was used, in the bottom row the SMILES generator was used.

the optimization of the molecular SMILES/SELFIES VAE toward high affinity to ABL1 and MAPK11 suggest that when comparing all molecules generated throughout the optimization process, the average pIC50 of the sampled molecules increased in all cases. Even though the improvement is mild in some cases, it can clearly be seen that the mean pIC50 of the distributions improved. Interestingly, for these two protein targets it moreover seems that relying on active site rather than full sequences slightly accelerated the generation of molecules with high affinity However, the results for the four JAK targets reveal conflicting evidence on a comparison between full sequences and active sites (cf. Table 4.4).

Ligand repr.	Kinase repr.	Baseline	Optimized
SELFIES	Full seq.	6.55 ± 0.6	6.60 ± 0.5
	Active site	6.51 ± 0.6	6.59 ± 0.5
SMILES	Full seq.	6.51 ± 0.6	6.57 ± 0.6
	Active site	6.57 ± 0.6	6.60 ± 0.5

Table 4.4: Results of GP optimization. We show the average binding affinity (pIC50) across the six kinases and all the molecules generated through the optimization process.

This is not a surprising finding given that the kinase representation is only used in the affinity evaluation and thus does not directly impact the molecular generation.

However, a tangible difference between the active site and full sequence model could still be found: the active site model saves computational resources because the models require less parameters and the average sequence length is smaller and so, training and, especially, inference speed are higher (cf. Table 4.5). Taken together with the better per-

Model	Config	# AAs	Model size	Inference time
k-NN	Full seq.	742 ± 369	–	59 ± 19ms
	Active site	29 ± 0	–	29 ± 8ms
BiMCA	Full seq.	742 ± 369	14.2M	18 ± 1ms
	Active site	29 ± 0	0.6M	6 ± 1ms

Table 4.5: Comparison of active site and full sequence models. Inference time was measured on 2.7 GHz Quad-Core `Intel Core i7`. The BiMCA affinity predictor was trained on a single `NVIDIA Tesla P100`.

formance of the active site models this suggests that it will, in almost all cases, be beneficial to rely on active site rather than full sequences (the sequence length is only 4% of the full protein and the model size is reduced by 25% Due to the fact that the BiMCA operates on batches of size 128, the inference time is still below the k-NN even when used on CPU. This is because the k-NN has to exhaustively compute distances for each sample and thus inference speed scales linearly with training data size. While the k-NN inference

speed linearly depends on the kinase sequence length and thus increases from ∼30ms to ∼ 100ms for the shortest (29) to the longest (2527) sequences in the dataset.

A positive side effect of the 25% speedup in inference runtime is that the convergence can be reached faster and the number of ligands sampled per time period can be increased, as shown in Table 4.6.

Evaluation	Active site	Sequence
Time until 5% pIC50 improvement (min.)	14 ± 8	21 ± 8
Number of effective ligands in 25mins	35 ± 19	30 ± 16

Table 4.6: Runtime comparison in sampling effective ligands. All ligands with a predicted IC50 $<$ 100nM (i.e., pIC50 $>$ 7) are considered effective.

4.6.2.1 Qualitative evaluation of molecules

In the top row of Figure 4.19 we display the ligands with the highest predicted affinity scores for each of the six targets and both kinase representations.

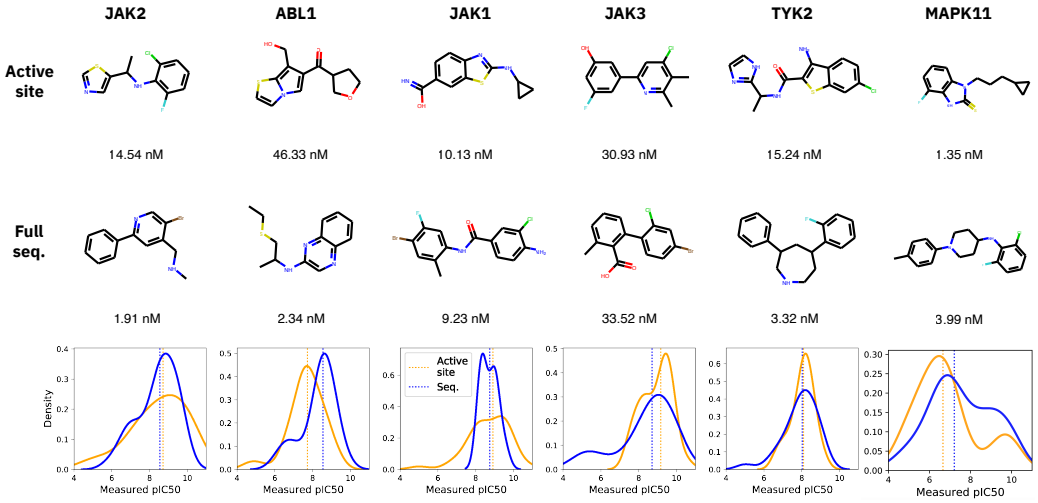


Figure 4.19: Best generated molecules per kinase. For each of the six kinases, the generated molecule with the lowest predicted IC50 value is shown. Instead, in the bottom row we display the distribution of *measured* pIC50 values of the most similar kinase inhibitors reported in BindingDB. Vertical dotted lines denote the medians.

This qualitative evaluation of the *de novo* kinase inhibitors reveals that the molecules are versatile in their structure and reach predicted affinities in the low nanomolar range ($2 - 50$nM). For each kinase, we selected the 5 most effective aromatic generated ligand and then retrieved the 10 most similar kinase inhibitors that were reported in BindingDB

(similarity measured with Tanimoto similarity on ECFP6 fingerprints). As can be seen in the bottom row of Figure 4.19, the distribution of measured pIC50 does not indicate any difference between full sequence and active site models.

4.6.3 Discussion

In this section we showed how Bayesian optimization techniques such as Gaussian Processes can be used to explore the latent space of molecular generative models in the search of novel kinase inhibitors. Using two generative models based on SMILES and SELFIES respectively, we find that the GP optimization successfully navigates the latent space towards regions of molecules that are predicted to bind better to kinase inhibitors. Moreover, we report that the (condensed) active site sequences can speed up the generation of binding ligands without sacrificing performance.

This section focused on single-property optimization but our GP approach is easily extendable to multi-objective optimization. While most previous work on GPs, however, focused on single-property optimizations [11, 320], multiobjective optimization has been tackled comprehensively with particle swarm optimization [324].

5 Bridging property prediction and conditional generation

In Part I of this thesis we have developed language models for molecular property and molecular interaction prediction. In Chapter 4 of Part II we then demonstrated how such models can *steer* conditional generative models to transform a biochemical context (e.g., a protein target sequence) into novel molecules with a high affinity to that context.

In this chapter, the last one of this thesis, we aim to *bridge* molecular property prediction and conditional molecular generative models through a multitask model, the Regression Transformer. This is motivated by three observations:

1. the controllability of molecular generative models has remained challenging thus far,
2. current molecular generative models lack an inductive bias that reflects *continuous* properties of interest, and
3. our previously developed methods are *global* search methods in the chemical space and can hardly be constrained on substructures to perform *local* search.

We will first reformulate regression as a conditional sequence modeling problem and then derive novel training objectives that are alternated during training to yield a multitask model that concurrently excels at regression and property-driven conditional generation tasks. We demonstrate that, despite using a nominal-scale training objective, the Regression Transformer matches or surpasses the performance of conventional regression models in property prediction tasks of small molecules, proteins and chemical reactions. Critically, priming the same model with continuous properties yields a highly competitive conditional generative model that can even outperform specialized approaches in a substructure-constrained, property-driven molecule generation benchmark.

The Regression Transformer thus forms a novel type of molecular generative model that can be simultaneously conditioned on arbitrary molecular substructures and continuous target properties of interest. This finds particular application in property-driven, local exploration of the chemical or protein space and could pave the road toward foundation models in material design.

5.1 On the need of unification

5.1.1 A next step in relaxing inductive biases

Transformers [50] are now ubiquitous in natural language processing (NLP) and have also enjoyed large success in molecular [28, 29, 325] and protein language modeling [326, 327]. The invention of Transformers was in alignment with the steady decline of inductive biases in ML, a trend that started with the rise of deep learning: CNNs outperformed traditional feature descriptors in object recognition [328], self-attention generalized dense layers to learn sample-dependent instead of static affine transformations [329] and Transformers exploited self-attention to supersede RNNs as the de-facto standard in NLP. The success of vision transformers has questioned the need for translation equivariance in image processing [330] and now, even frozen Transformers pretrained on text achieve state-of-the-art results in object detection and protein classification [331]. Given that Transformers are today's most generic deep learning model[1], it is not surprising that attempts have been made to abstract entire domains like RL to sequence modeling in order to leverage Transformers [332].

In classical regression tasks, however, the impact of Transformers has been restricted to large-scale unsupervised pretraining followed by finetuning a task-specific regression head [333, 334, 335]. This procedure still relies on the conventional approach of modeling target variables (i.e., properties) as functions of input variables (i.e., tokenized sequences, most typically natural text). A provocative next step toward reducing inductive biases might be to refrain from following this classic, discriminative modelling approach and rather learn the joint distribution over input and target variables. This could effectively further blur the lines between predictive and conditional generative models. The feasibility of such approach can be assessed via permutation language modeling (PLM), an extension of masked-language-modeling to autoregressive models [336].

5.1.2 Implications for molecular modeling

Such dichotomous models (that concurrently excel at regression and conditional sequence generation) are beyond applications in NLP of special interest for chemical and material design. Molecules are often labelled with continuous properties (e.g., drug efficacy or protein solubility) and design tasks are intertwined with bio- or physicochemical properties. But despite the rise of deep generative models in molecular [337, 338] and protein design [339, 340], current approaches still develop property predictors and generative models independently. Transformer-based architectures have been used widely on chemical tasks but either focused on property prediction [341, 342] or on conditional molecular design [343, 344], never on both. This semantic gap persists across architectural flavors (e.g., GANs [221], RL [345], VAEs [11],

[1] graph neural networks with multihead attention as neighborhood aggregation on complete graphs.

GNNs [344, 346], flow [24, 207] and diffusion models [208]). To our knowledge, all existing approaches either tune task-specific heads [335] or limit the communication between both modules to a reward and thus fail to "entangle" constrained structure generation with property prediction. This critically violates the intuitive expectation that a property-driven generative model should, in the first place, excel at recognizing this property.

5.1.3 Scope of the chapter

In this chapter, we aim to close this gap by reformulating regression as a sequence modeling task. We propose the Regression Transformer (RT), a novel multitask model that can be trained on combinations of numerical and textual tokens (see Figure 5.1).

This circumvents the canonical way of addressing regression in Transformers, i.e., tuning a designated regression head [334]. Despite solely relying on tokenization of numbers and cross-entropy loss, the RT can successfully solve regression tasks. Notably, the same model can conditionally generate text sequences given continuous properties. This is achieved simply by moving the `[MASK]` location and does not require finetuning specific heads; thus constituting a true multitask model. To equip the RT with an inductive bias for handling floating-point properties, numbers are first tokenized into a sequence of tokens preserving the decimal order. We then devise numerical encodings to inform the model about the semantic proximity of these tokens. To allow for concurrent optimization of regression and conditional generation, we derive a PLM-inspired, alternating training scheme that includes a novel self-consistency loss for improved text generation based on continuous primers.

We will describe and assess the capabilities of the RT on a diverse set of predictive and generative tasks in chemical and protein language modeling. We commence with small-molecule modeling, validate the RT on a synthetic dataset of drug-likeness [248] and then test it on three property prediction datasets from the MoleculeNet benchmark [34]. The property predictions results are compared with previous approaches relying on a regression loss and demonstrate that regression can be cast as conditional sequence generation task without losing accuracy. Although we aim to concurrently excel at predicting properties and generating sequences conditioned on properties, we start training with the PLM objective [336] which does not explicitly model those tasks. We then refine this objective and devise a training scheme that alternates between optimizing property prediction and text generation. For the latter, we derive a novel self-consistency loss that exploits the dichotomy of the RT by querying itself with the generated candidate sequence. To assess performance in conditional sequence generation, we systematically vary the continuous properties of interest and investigate the model's ability to adapt a seed sequence according to the primed property value. We show applications on property-driven local chemical space exploration by decorating scaffolds with a continuum of properties and evaluate the novel molecules using the RT itself as well as an independent property pre-

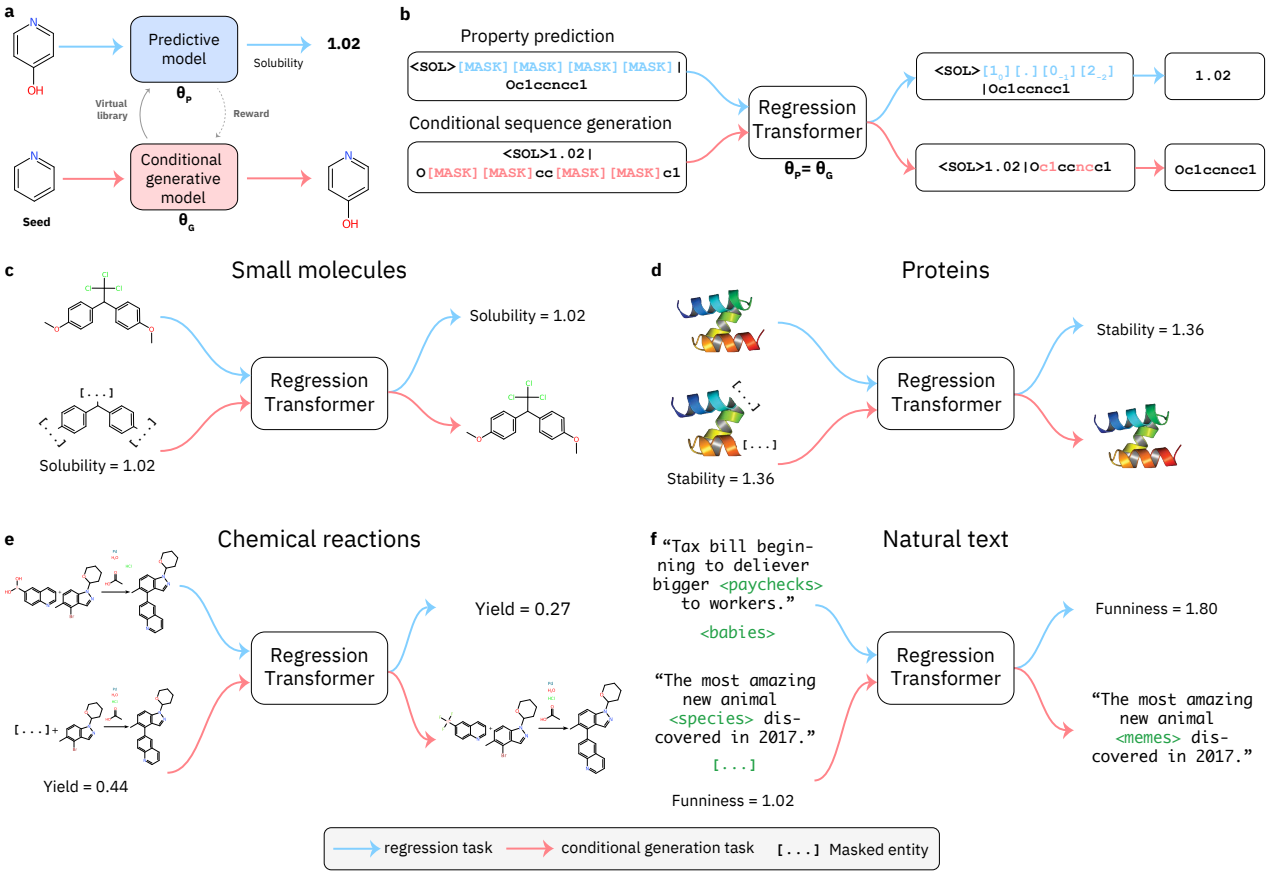


Figure 5.1: Overview of Regression Transformer (RT). The RT is a multitask language model designed to handle combinations of text and numbers. **a)** Traditional approach in generative chemistry: property predictors and generative models are trained independently from another. **b)** Our approach: Training the RT yields a dichotomous model that seamlessly switches between property prediction and conditional text generation. The model's task is to fill the content behind the `[MASK]` tokens. Depending on the mask location, the same model either predicts numerical tokens given textual tokens, thus performing a regression task *(blue stream, top)*; or predicts textual tokens given both numerical and textual tokens, thus performing a property-driven conditional generation *(yellow stream, bottom)*. **c) - f)**: This novel formulation finds application across a wide range of domains. We demonstrate the flexibility of the RT in predictive and generative tasks in modeling small molecules, proteins, chemical reactions. It can even be applied to natural text.

dictor [36]. The RT is then challenged against specialized molecular generative models on a property-driven molecular generation benchmark [19], where it significantly outperforms prior art.

Next, the RT is investigated on protein sequence modeling where it matches the performance of conventional Transformers on two regression datasets from TAPE [333]. In

experiments on chemical reactions, we notice that the RT constitutes a generalization of forward reaction and retrosynthesis models. We then demonstrate on two reaction datasets that the RT can not only predict reaction yields with similar accuracy to conventional Transformers [62], but that it can also substitute specific precursors and thus generate novel reactions with higher predicted yield than a seed reaction. Last, we apply our proposed methodology to an NLP task where we find that it can benefit sequence prediction tasks.

5.1.4 Structure-constrained molecular generation

Most molecular generative models mentioned throughout this thesis as well as those methods developed above search *globally* in the chemical space. In molecular discovery applications, however, the design process oftentimes does not start from scratch but from an existing molecule, e.g., a hit compound identified in a HTS. For such cases, methods that locally search the chemical space to optimize desired properties are needed. In an early study, *Arús-Pous* et al. [347] proposed the deep scaffold decorator which exhaustively slices molecules into scaffolds and decorations and then samples decoys to decorate the scaffold, thus locally explore the chemical space around a molecule. Others used graph-based generative models [346, 348, 349, 350, 351, 352, 353] which can also be tuned to optimize physicochemical properties. However, apart from the work by *Li* et al. [349], none of the methods can be conditioned on a desired property at inference time. Even *Li* et al. [349] could not *jointly* condition on a desired property and a scaffold. Among others, this task will be addressed in depth in this chapter with the Regression Transformer.

5.2 The Regression Transformer

5.2.1 XLNet backbone – Unifying language modeling paradigms

The Regression Transformer (RT) is built upon an XLNet backbone [336] to retain the benefits of auto-regressive modeling in combination with a bidirectional context. Essentially, XLNet unifies the two main paradigms for pretraining language models: Auto-encoding and auto-regressive. In conventional language models we maximize the likelihood with an auto-regressive (AR) factorization

$$\max_{\theta} \sum_{t=1}^{T} \log p_\theta(x_t|\mathbf{x}_{<t}) \tag{5.1}$$

where $\mathbf{x} = [x_1, ..., x_t]$ is our text sequence and p_θ is learned by the language model. The disadvantage of AR modeling is the unidirectional information flow that conflicts with many downstream tasks. For example, for the RT a bidirectional context will be critical because SMILES (or SELFIES) strings are non-local sequences. Masking functional groups usually implies masking disconnected tokens, hence we wish to have a model that can fill multiple tokens at arbitrary positions in a sequence while attending the full remaining sequence. This can naturally be achieved within the second paradigm – auto-encoding based pretraining of language models. This is a self-supervised setting, originally introduced by *Devlin* et al. [334] to train BERT. It aims to reconstruct the original sequence from a corrupted sequence. Let $\hat{\mathbf{x}}$ be a corrupted version of $\mathbf{x}$ where a fraction of the tokens is masked, then the objective of BERT-based auto-encoding pretraining becomes:

$$\max_{\theta} \sum_{t=1}^{T} m_t \log p_\theta(x_t|\hat{\mathbf{x}}) \tag{5.2}$$

where $\mathbf{m}$ is the masking indicator vector. This is a powerful pretraining objective, however the critical disadvantage is the independence assumption which assumes that all corrupted tokens can be reconstructed independently. This becomes increasingly disruptive as more masked tokens are filled, hence BERT-based models do not perform well at generating longer sequences. This limits BERT's applicability for generative tasks in biochemistry like scaffold decoration where large portions of a molecule might be masked and generation of individual atoms can critically alter the molecule's functional properties. The contribution of *Yang* et al. [336] in XLNet is to unify both paradigms and propose an auto-regressive language model that yields, in expectation, full bidirectional attention.

In the next subsection, we will first recapitulate the original XLNet objective and then extend it to reformulate regression as a conditional sequence modeling task. In general, it is important to notice that our proposed framework can be applied to all Transformer flavors, but it certainly benefits from an autoregressive generation with full sequence attention even for discontiguous mask locations, like XLNet or MPNet [354].

5.2.2 Reformulating regression as conditional sequence modeling task

This section describes the training objectives for the RT. The input $\mathbf{x}$ for a RT is defined by a concatenation of k property tokens $[\mathbf{x}^p]_k$ and l textual tokens $[\mathbf{x}^t]_l$, such that: $\mathbf{x} = [\mathbf{x^p}, \mathbf{x^t}]_T = [x_1^p, ..., x_k^p, x_1^t, ..., x_l^t]_T$. The full sequence length is $T{=}k{+}l$ and $\mathbf{x}^p$ and $\mathbf{x}^t$ are property and textual tokens respectively.

Permutation language modeling (PLM) objective. The idea of PLM [336] is to fill multiple masked tokens auto-regressively by sampling a factorization order $\mathbf{z}$ for a sequence $\mathbf{x}$ at runtime. Decomposing the likelihood $p_\theta(\mathbf{x})$ according to the factorization

order yields, in expectation, a bidirectional auto-regressive model. Let $\mathbf{z} \in \mathcal{Z}_T$ denote one of the $T!$ permutations of our sequence $\mathbf{x}$. If z_i and $\mathbf{z}_{<i}$ are the i-th and first $i-1$ elements of $\mathbf{z}$, the PLM objective is:

$$\max_{\theta} \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{i=1}^{T} \log p_\theta(x_{z_i}|\mathbf{x}_{\mathbf{z}_{<i}})\right] \tag{5.3}$$

In practice, partial prediction is performed, i.e., only the last c tokens of the factorization order $\mathbf{z}$ are predicted. Following XLNet, $\mathbf{z}$ is split into a (masked) target subsequence $\mathbf{z}_{>c}$ and an unmasked input sequence $\mathbf{z}_{\leq c}$ s.t. the objective becomes

$$\begin{aligned}\mathcal{J}_{PLM} &= \max_{\theta} \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\log p_\theta(\mathbf{x}_{\mathbf{z}>c}|\mathbf{x}_{\mathbf{z}_{\leq c}})\right] \\ &= \max_{\theta} \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{i=c+1}^{T} \log p_\theta(x_{z_i}|\mathbf{x}_{\mathbf{z}_{<i}})\right]\end{aligned} \tag{5.4}$$

where c is a hyperparameter, usually sampled per batch such that the fraction of masked tokens is roughly $1/c$. We notice that Equation 5.4 does not make any specific choices on $\mathbf{x}^p$ and $\mathbf{x}^t$. It thus constitutes our baseline objective. While Equation 5.4 is a generic objective, it is computationally exhaustive to optimize due to the permutations. Moreover it is not ideal for our needs because it does not distinguish between textual and property tokens. Instead, we are aiming to develop a single model that can either predict numerical tokens (when given text sequences) or text tokens (when given a combination of numerical and text tokens). To that end, we propose to train on two alternating objectives, one designed for property prediction and one for text generation.

Property prediction objective. Instead of randomizing which tokens are masked, this objective exclusively masks all the property tokens. Specifically, we constrain the factorization order $\mathbf{z}$ by setting the first l elements to $\mathbf{x}^{\mathbf{t}}$ and fixing $c = l$. This guarantees that only property tokens are masked. Let $\mathcal{Z}_T^p$ denote the set of possible permutations. Under this constraint, then the objective becomes

$$\begin{aligned}\mathcal{J}_P &= \max_{\theta} \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T^p}\left[\log p_\theta(\mathbf{x}^p|\mathbf{x}^t)\right] \\ &= \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T^p}\left[\sum_{i=c+1}^{T} \log p_\theta(x^p_{z_i}|\mathbf{x}^t_{\mathbf{z}_{\leq c}}, \mathbf{x}^p_{\mathbf{z}_{>c<i}})\right]\end{aligned} \tag{5.5}$$

where $\mathbf{x}^p_{\mathbf{z}_{>c<i}}$ denotes the c-th to the $i-1$-th element of the factorization order $\mathbf{z}$. We emphasize that this "tailored" property objective $\mathcal{J}_p$ is still optimized with a cross-entropy loss in practice. Note that this loss cannot convey any notion on the qualitative proximity of the prediction to the labels because the level of measurement of tokens in a language

model are on a nominal level. Thus, predicting a sequence of numerical tokens corresponding to a property score of 0.91 for a sample with a true property of 0.11 will not generally result in a higher loss than predicting 0.21. Instead, a traditional regression loss operates on a ratio scale.

Conditional text generation objective. This objective facilitates the generation of textual tokens given a property primer and textual tokens. We constrain the factorization order $\mathbf{z}$ by setting the first k elements to $\mathbf{x}^p$ to and sampling the cutoff c, s.t. $c \leq k$. This ensures that masking only occurs on textual tokens. With this constraint, we denote the set of permutations by $\mathcal{Z}_T^t$ and the objective becomes

$$\begin{aligned} \mathcal{J}_G &= \max_{\theta} \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T^t}\left[\log p_\theta(\mathbf{x}^t_{\mathbf{z}_{>c}}|\mathbf{x}^p_{\mathbf{z}_{\leq k}}, \mathbf{x}^t_{\mathbf{z}_{>k<c}})\right] \\ &= \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T^t}\left[\sum_{i=c+1}^{T} \log p_\theta(x^t_{z_i}|\mathbf{x}^p_{\mathbf{z}_{\leq k}}, \mathbf{x}^t_{\mathbf{z}_{>k<i}})\right] \end{aligned} \tag{5.6}$$

Intuitively, this objective applies regular PLM while sparing the numerical tokens. It then aims to reconstruct the full text sequence (i.e., molecule) given the uncorrupted property tokens and partially corrupted textual tokens.

Self-consistency objective. Standalone, the above conditional text generation objective (Equation 5.6) does not reward if the generated sequences adhere to the primed property. It only rewards the accuracy of sequence reconstruction. However, in chemical as well as natural languages, changes in single tokens (i.e., atoms, amino acids or (sub)words) can drastically change the property (meaning) of a molecule (sentence). To reward the model for generating molecules that differ from the seed sequence but whose property value follows the primed property, we extended the text generation objective $\mathcal{J}_G$ by a self-consistency term that exploits the dichotomy of the Regression Transformer. The full objective is given by:

$$\mathcal{J}_{SC} = \mathcal{J}_G(\mathbf{x}) + \alpha \cdot \mathcal{J}_P(\hat{\mathbf{x}}) \tag{5.7}$$

where the second addend is the self-consistency term, weighted by a factor α. Intuitively, it is given by the difference between the property of the sample and the predicted property of the generated sample $\hat{\mathbf{x}}$. Here, $\hat{\mathbf{x}}$ is obtained by greedy decoding of the masked tokens and combining it with the non-corrupted tokens of $\mathbf{x}$. To be precise, $\hat{\mathbf{x}} = [\mathbf{x}^p, \hat{\mathbf{x}}^t]$ where $\hat{\mathbf{x}}^t = [m_1\bar{x}_1 + (1-m_i)x_1, ..., m_l\bar{x}_l + (1-m_l)x_l]$. Here, $\mathbf{m}$ is an indicator vector whether masking occurred at a given position and $\bar{\mathbf{x}} = [\bar{x}_1, ..., \bar{x}_l]$ is the result of greedy decoding. In such a formulation, the RT acts as an oracle during its own optimization, resembling an additional layer of self-supervision. This provides a notion of self-consistency to the model in that it rewards the generation of *any* molecule that adheres to the primed prop-

erty. Without such a loss, the model would be perform a pure reconstruction that does not emulate the flexibility and versatility of a property-driven generative models that is desired at inference time. The disadvantage of this loss is that it may induce side effects in situations where the model struggles to learn the property prediction.

5.2.3 Tokenization

This section describes the processing of alphanumeric sequences, i.e., strings consisting of a mixture of numerical and textual symbols. This process is visualized in Figure 5.2 (*top*). Unlike previous approaches that modelled 8-bit integers such as pixels with a clas-

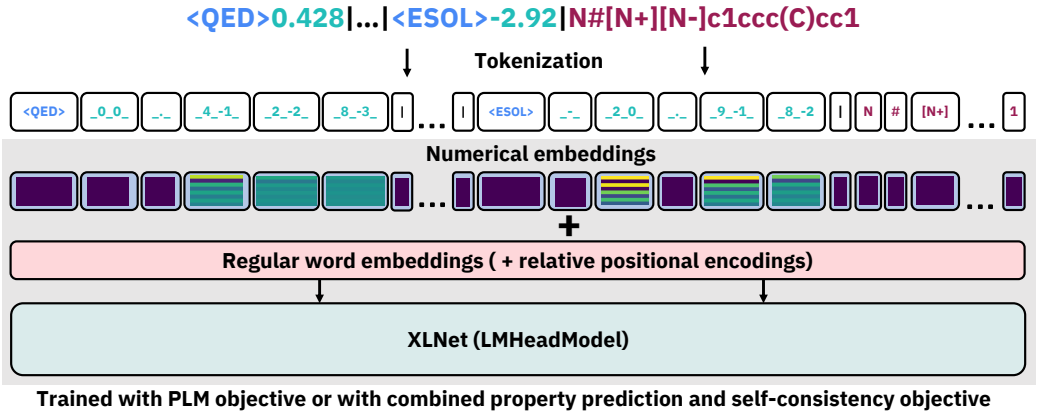


Figure 5.2: Workflow of the Regression Transformer (RT) model. Based on the XLNet backbone, the RT is a dichotomous model designed to handle combinations of text and numbers. *Top:* An input sequence consisting of a molecular string (red) and two property tags (blue), each associated to a floating value (green). Numbers are tokenized into a sequence of tokens that preserve the decimal order of each character. The pipe (|) is a separator token distinguishing numerical and text tokens. *Middle*: We propose numerical encodings that inform the model about the semantic proximity of these tokens and naturally integrate with relative positional encodings and classical learned embeddings. *Bottom:* The RT is trained with an alternating training scheme, derived from the PLM objective [336] and designed to concurrently optimize property prediction and conditional generation (*bottom*). The dots indicate that the RT naturally scales to multiple property tags.

sifier [355], we strive to represent real numbers with arbitrary floating point precision. Since representing every number as a single token is suboptimal due to a lack of generalization to new numbers and sparsity of the provided tokens, we formulate regression as sequential categorical task. In turn, this necessitates a scheme for converting text repre-

senting numbers into a sequences of tokens. First, the following regular expression splits a string denoting a numerical:

```
\s*\s*?(\+|-)?(\d+)(\.)?(\d+)?\s*
```
(5.8)

Each of the resulting matches containing a number is converted to a token $t_{v,p}$ where $v \in \mathbb{N} \cap [0..9]$ is the value/digit and $p \in \mathbb{Z}$ is the decimal place (e.g., 12.3 is split into [`1_1`, `2_0`, `.`, `3_-1`]). We call these *numerical tokens*. This representation has the advantage that it allows easy decoding of the digit sequence but also distinguishes their decimal order by adhering to classic positional notation. Negative numbers are preceded with a special token. Regarding alphabetic tokens, we represent molecules as SELFIES [65] strings and tokenized them with their internal tokenizer. In one ablation study, we instead use SMILES [48] and tokenize with the regular expression from *Schwaller* et al. [53]. Protein sequences are tokenized per amino acid.

5.2.4 Numerical encodings

5.2.4.1 Float encodings

The inherent structure of numbers conflicts with the notion of learned embeddings in language models. In other words, having to learn the embeddings of tokens corresponding to *numbers* might be tremendously ineffective. Remember that the RT is trained with a normal cross-entropy objective which implies that a notion of similarity between numerical tokens can not be conveyed. As a remedy, we devised what we call "numerical encodings" (NE); a simple inductive bias that informs the model about semantic proximity of numerical tokens, similar to the positional encodings proposed by *Vaswani* et al. [50]. The objective of these encodings is to circumvent the necessity of having to learn from scratch the semantic similarities between the different numerical tokens. In practice, we sum the NEs with regular word embeddings and relative positional encodings from XL-Net (see Figure 5.2 for the workflow). The NEs are zero vectors for all but numerical tokens. We follow positional notation as above. Given a token $t_{v,p}$ (with digit value v and decimal place p), the numerical encoding at embedding dimension j is defined as:

$$NE_{Float}(v, p, j) = (-1)^j \cdot \frac{v \cdot 10^p}{j+1} \tag{5.9}$$

Thus, the amplitude of the NE scales with the numerical value of the token. The NEs are perfectly correlated among the embedding dimensions. For even and odd dimensions, they alternate between positive and negative values. In addition, they vanish for higher dimensions (see example in Figure 5.3**a**). Importantly, the NEs were devised in such a way that their pairwise distances are symmetric and decay monotonically with the float value (see Figure 5.3**b**).

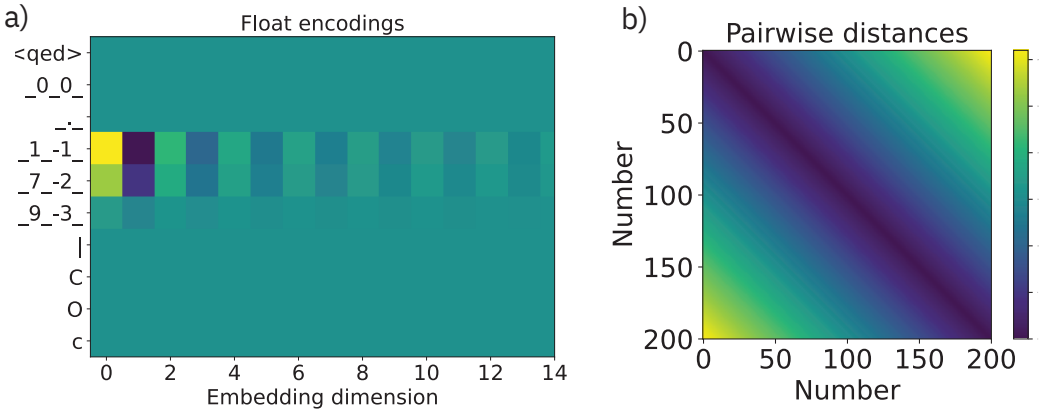


Figure 5.3: Float-based numerical encodings. a) Numerical encodings for an molecule with a QED of 0.179. **b)** Pairwise distances of numerical encodings for floats between 0 and 100 (the NEs of all tokens associated to a float are summed up).

5.2.4.2 Integer encodings.

As an alternative to the float-based numerical encodings (NE), we experimented with an encoding scheme relying solely on positive integers. Note that any regression problem can trivially be casted to a regression problem where all labels are positive integers. Under this consideration, we need to define NEs only for positive integers[2]; similar to positional encodings. We therefore propose to directly utilize the definition from *Vaswani* et al. [50] as NEs:

$$\begin{aligned} NE_{Int}(v, p, 2j) &= \sin\left[(v \cdot 10^p)/10000^{2j/d_e}\right] \\ NE_{Int}(v, p, 2j{+}1) &= \cos\left[(v \cdot 10^p)/10000^{2j/d_e}\right] \end{aligned} \tag{5.10}$$

where d_e is the embedding size. The advantage of this integer-based encoding is that every embedding dimension captures fluctuations of different frequencies; using trigonometric functions as continuous analogs to alternating bits. Practically, to use the Integer-NEs, the property values were casted to the range $[0, 1000]$ and rounded.

5.2.5 Training & evaluation procedure.

Due to the amount of experiments and investigated datasets in this chapter, we describe data processing and training procedures in detail within each section. Here we only list the generic things that applied to all experiments equally.

All experiments build upon the XLNet [336] backbone from the `HuggingFace` library [356]. As visualized in Figure 5.2, we expanded the XLNet backbone with

[2] Strictly speaking only integers with a single, non-zero digit (i.e., covered by the base-10 exponentiation of the decimal system)

our proposed tokenization scheme (Subsection 5.2.3), an additional encoding layer for the numerical embeddings with $N_{dim} = 16$ (Subsection 5.2.4 and the custom training objectives (Subsection 5.2.2). Regarding architectural hyperparameters, we used 32 hidden layers in the Transformer encoder, with a dimensionality of 256 and 1024 in the feed-forward layer and 16 attention heads (20% dropout). Altogether, this model has $\sim$ 27M trainable parameters (exact numbers vary dependent on vocabulary size). During evaluation, greedy decoding was used for property prediction and beam search decoding for conditional sequence generation. We used `PyTorch` 1.3.1 [159] and `Transformers` 3.1.0 [356]. All models were trained on single GPUs (`NVIDIA Tesla A100` or `V100`).

Regression.

To solve regression (or property prediction) tasks, we convert the sequence of predicted (numerical) tokens into a floating-point prediction. Note that the model never failed to predict a token sequence corresponding to a valid numerical. We then report the root-mean-squared error (**RMSE**), Pearson's correlation coefficient (**PCC**) or the coefficient of determination ($\mathbf{R^2}$), dependent on the dataset and previous methods.

Conditional sequence generation.

Dependent on the application domain, different metrics are utilized. Details will follow in the respective sections.

k-NN baseline model

Besides comparing our results to previously published work, we also compared the molecular property prediction results to a simple k-NN baseline model. The distance measure was (inverted) Tanimoto similarity [162] of ECFP4 fingerprints [46]. For the protein language models, the Levenshtein distance between the protein sequences was used [161]. For the k-nn baseline models, k was determined based on the best performance on the validation data. This led to $k = 25$ for the drug-likeness/QED task, $k = 21$ for the protein interaction (Boman index) task, $k = 50$ for the fluorescence and $k = 15$ for the stability task.

5.3 Benchmarking a classifier against regression models

5.3.1 Data preparation and evaluation procedure

Drug-likeness (QED)

Dataset. Starting from $\sim$ 1.6M bioactive molecules from ChEMBL [357], we created a synthetic dataset by computing the QED [248] score (q $\in [0, 1]$) for all molecules with `RDKit` and rounded to 3 decimal places. We used $\sim$ 1.4M molecules for training, 1k for validation and 10k for testing.

Training procedure. We started training the models with the PLM objective (Equation 5.4) on the QED dataset until validation perplexity saturated ($\sim$ 4 days, single-GPU). Thereafter, the models were further refined on the same dataset by alternating every 50 steps between the property prediction objective (Equation 5.5) and the text generation objective (Equation 5.7). We perform ablation studies on the self-consistency loss, setting α in (Equation 5.7) to 0 and 1 respectively. During the latter, we gave the model more flexibility by setting $c = 2.5$, s.t., $\sim 40\%$ of the tokens were masked (maximum span: 7 tokens). The SELFIES/SMILES vocabulary had 509 and 724 tokens respectively.

MoleculeNet datasets.

Data. We focused on 3 regression datasets from the MoleculeNet benchmark [34]: *ESOL*, *FreeSolv* and *Lipophilicity*, where the task is to predict water solubility, hydration free energy and lipophilicity of a molecule, respectively. For each dataset, we performed 3 random splits (as recommended by [34]) with 15% validation data. Because the datasets are small (< 5000 samples), we used SMILES augmentation [56] to augment the dataset by a factor of 16.

Training procedure. For the MoleculeNet datasets, the models were warm-started using the QED initialization and trained only for 50k steps (batch size 4) with early stopping. Since the QED pretraining utilized numerical values in $[0, 1]$, we normalized the regression values of the MoleculeNet datasets to the same range and rounded them also to three decimal places. For all objectives, unless otherwise constrained, we set the masking hyperparameter $c = 5$ and restrict the span of consecutively masked tokens to a maximum of 5 tokens.

Evaluating molecular generation

For all these small molecule datasets, we strive to assess the model's ability to decorate an arbitrary, possibly discontiguous fractional input sequence (e.g., a molecular scaffold) according to a property of interest. Therefore, we randomly mask a fraction of tokens of the text sequence and then query the model with ten equidistant property primers spanning the full range of property values. We decode ten molecules and the reported metric is the average **Spearman's ρ** between the primers and the actual property values (after removing duplicate molecules). Spearman is favorable over Pearson because it is only rank-sensitive. Note that due to constraints induced by the fragmented sequence, covering the entire property spectrum is usually impossible such that e.g., RMSE is inappropriate for this task (e.g., priming a highly toxic scaffold with low toxicity cannot yield a non-toxic molecule). As a sanity check, we also report "Zero-Variance" (**0-Var**), i.e., the percentage of samples for which the generation was unaffected by the primer (the lower the better).

5.3.2 Initial validations – learning drug-likeness

5.3.2.1 Permutation language modeling training

To test the feasibility of concurrent property prediction and conditional generation, we start with optimizing the vanilla permutation language objective (Equation 5.3) on a synthetic QED dataset (see Figure 5.2 for an illustration of how the mixed alphanumeric sequences are tokenized and embedded). Since this objective masks tokens randomly in the sequence, evaluating such models on property prediction (i.e., masking only numerical tokens; cf. Figure 5.1**b** *top*) does not closely mimic their training dynamics.

As can be seen Table 5.1, despite this mismatch between training and evaluation (as well as the unconventional formulation of a regression task as sequence modeling), all models generated sequences of numerical tokens that allowed decoding floats, and even achieved a RMSE < 0.06. In this setting, from the two types of proposed numerical encodings, the float-based encodings yielded slightly superior result to integer-based encodings.

Instead, for the generative task, the same models were queried 10 times for every validation molecule with property primers[3] equidistantly spaced in $[0, 1]$ and 40% of masked textual tokens. The high rank correlation ρ (between primers and QED of unique, generated molecules) values show that the model learned successfully to complete the corrupted sequences to produce full molecules with a desired QED. Here, the SELFIES models exceeded the SMILES models by far, because SMILES, unlike SELFIES, can be syntactically invalid. Due to the comparable results for property prediction, the remaining experiments focus exclusively on SELFIES. Notably, the novelty score (i.e., percent-

[3] Throughout this chapter by "primers" we mean that we replace the true property of a sequence with a desired property value.

Configuration			*Regression task*		*Generation task*	
Data	NE	Perplexity (↓)	RMSE (↓)	PCC (↑)	0-Var (↓)	SCC (↑)
SMILES	–	$\mathbf{1.55}_{\pm 0.02}$	$0.0549_{\pm 0.01}$	$\mathbf{0.972}_{\pm 0.01}$	$1.6\%_{\pm 0.2}$	$0.096_{\pm 0.02}$
SELFIES	–	$1.61_{\pm 0.03}$	$0.0591_{\pm 0.00}$	$0.968_{\pm 0.00}$	$0.9\%_{\pm 0.2}$	$0.427_{\pm 0.01}$
SELFIES	FE	$1.59_{\pm 0.03}$	$\mathbf{0.0547}_{\pm 0.01}$	$0.971_{\pm 0.00}$	$\mathbf{0.3}\%_{\pm 0.1}$	$\mathbf{0.467}_{\pm 0.01}$
SELFIES	Int	$1.63_{\pm 0.02}$	$0.0564_{\pm 0.00}$	$0.968_{\pm 0.00}$	$0.8\%_{\pm 0.3}$	$0.440_{\pm 0.01}$

Table 5.1: Performance after PLM training. RMSE (↓) and PCC (Pearson correlation coefficient) refer to predicting QED, perplexity (↓) to the PLM objective (Equation 5.4) and Spearman ρ (↑) and 0-Var (↓) to the conditional generation task. All values are means across multiple models. All numbers computed on 10k test samples. NE refers to the use of numerical encodings. FE refers our Float-based numerical encodings whereas "Int" refers to the Integer-based numerical encodings.

age of conditionally generated molecules not present in training data) was $> 99\%$ for all models. This demonstrates that the RT can generate novel chemical matter that adheres to a continuous property of interest. Moreover, the numerical encodings (NE) slightly improved performance in all tasks.

5.3.2.2 Ablation study on numerical encodings

In this subsection, we will examine different types of numerical encodings more closely and justify our choices for further experiments. Table 5.1 showed that the effect of the type of numerical encoding (float-based or integer-based) seems to be minor. For that experiment, we followed the common approach of *summing* these encodings to the normal, learned embeddings. This is the most common approach in the literature [50, 336] even though we note that disentangling content and position embeddings can improve language models [358]. To test this effect, we conducted an ablation study where we, instead of summing the numerical encodings to the regular embeddings, we concatenated them (dimensionality of 32 for the NEs.). The results in Table 5.2 reveal slightly inferior but nearly identical results.

NE	Type	RMSE (↓)	PCC (↑)
–	–	$0.0591_{\pm 0.00}$	$0.968_{\pm 0.00}$
Float	Concat.	$0.0581_{\pm 0.00}$	$0.966_{\pm 0.01}$
Float	Sum	$\mathbf{0.0547}_{\pm 0.01}$	$\mathbf{0.971}_{\pm 0.00}$
Int	Concat.	$0.0666_{\pm 0.01}$	$0.963_{\pm 0.01}$
Int	Sum	$0.0564_{\pm 0.00}$	$0.968_{\pm 0.00}$

Table 5.2: Ablation study on NEs. Results on PLM training.

For the rest of our work, we decided to use a summation for three reasons:

1. It avoids any additional hyperparameters and model weights.
2. Using a summation probably still yields approximately orthogonal subspaces of token embedding and numerical encodings (due to the high dimensionality). Hence, the curse of dimensionality might obviate the need to enforce orthogonality with a concatenation.
3. Only the float-based rather than the integer-based encodings can be applied to floating numbers (i.e., for experiments with integer-based encodings we always casted the property values to the range $[0, 1000]$ which is undesired in practice since it requires postprocessing).

While we conjectured that using NEs improves the performance in both tasks (property prediction and conditional generation), we emphasize that providing this prior might not be necessary given enough data. We hypothesize that refining our NEs might yield better results and in particular a faster convergence, but leave further refinement to future work, especially given the plethora of research about positional encodings [359, 360, 361].

5.3.2.3 Alternating training with refined objectives

Next, based on our proposed training scheme with alternating objectives, the models were refined: For every model in Table 5.1, two models were trained, *without* ($\alpha = 0$) and *with* ($\alpha = 1$) the self-consistency term in the text loss (cf. Equation 5.7), respectively. As shown in Table 5.3, the performance in regression as well as conditional generation improved significantly, demonstrating the effectiveness of the refined objectives. Moreover,

Configuration		*Regression task*		*Generation task*	
NE	α	RMSE	PCC	0-Var	Spearman ρ
✗	0	**0.0341**	**0.988**	0.2%	0.47
✗	1	0.0483	0.978	0.3%	0.49
✓	0	0.0498	0.982	0.3%	0.47
✓	1	0.0367	0.987	**0.2%**	**0.52**

Table 5.3: Performance evaluation on refined objectives. Legend like in Table 5.1. NE means numerical encodings and α refers to the self-consistency loss function in Equation 5.7. All models here used SELFIES.

all configurations of the Regression Transformer (RT) outperformed a baseline k-NN-regressor on Tanimoto similarity and our best configuration even surpassed the SMILES-BERT model [341] which achieved a MAE of 0.02 after pretraining on ∼9M SMILES with a regular regression loss (see Table 5.4). The self-consistency term further improved the model's ability to generate tailored ensembles of molecules and led to consistently higher correlation scores. Generally, the better performance of the self-consistency models ($\alpha = 1$) in the generative tasks comes at the cost of slightly inferior regression per-

Model	MAE (↓)
k-NN (baseline)	0.054
SMILES-BERT [342]	0.020
RT - PLM objective (Equation 5.3)	0.035
RT - Alternating objective (Equation 5.6)	**0.017**

Table 5.4: Performance comparison in predicting QED. MAE stands for mean absolute error. The RT with alternating objectives used $\alpha = 0$ in Equation 5.7.

formance (cf. Table 5.3). Presumably, this is because the model weights in charge of the regression are confounded with the gradients from the self-evaluation (cf. Equation 5.7). The novelty scores for the molecules generated in this setting were even slightly higher than for the PLM training ($> 99.3\%$ for all models).

5.3.2.4 Examples on molecule decoration

An example decoration is shown in Figure 5.4 (*top*) where a single seed molecule is decorated according to the property primers to cover the full range of QED scores. It can be seen that the model adapted the seed molecule based on the property primers in such a way that the generated molecules were largely consistent with the provided property primer.

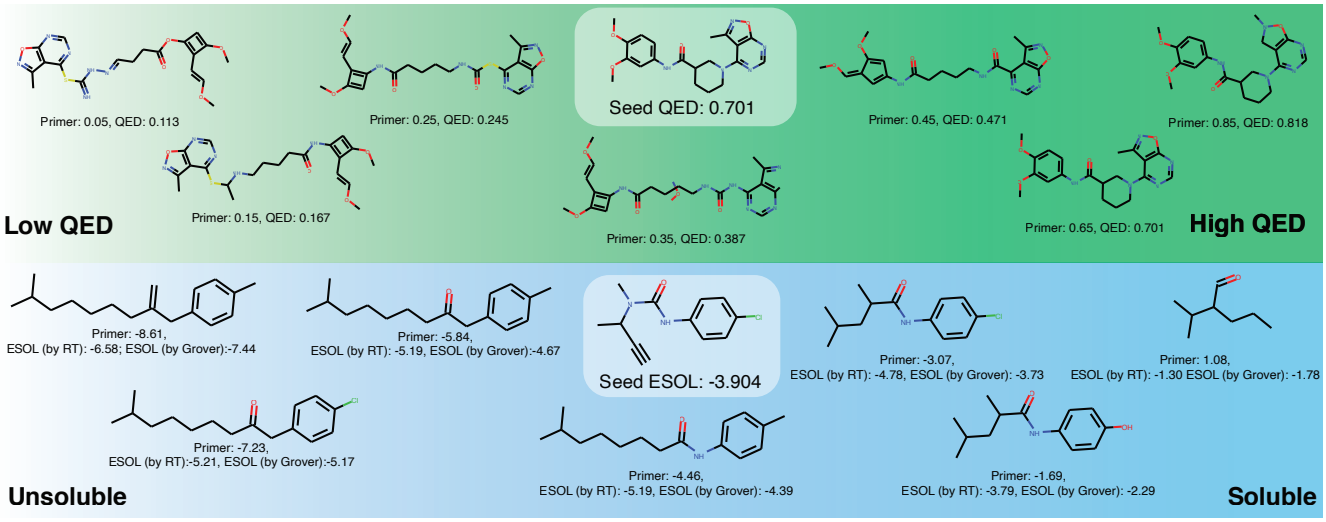


Figure 5.4: Property-driven, local chemical space exploration. For each row, the seed molecule is shown in the middle. Based on 10 property primers, 10 molecules were decoded (duplicates were discarded). *Top:* QED dataset. *Bottom:* ESOL dataset of aquatic solubility. The solubility was predicted by the RT itself but is also externally validated with predictions from Grover [36].

A particularly challenging application for property-driven, local exploration of the chemical space is scaffold decoration. This is a technique in medicinal chemistry with the goal to discover novel compounds by modifying the central core structure (i.e., removing substituents while retaining rings and their linker fragments) of known compounds [362]. We simulated this task on the QED dataset by determining the scaffold with `RDKit` and masking only the non-scaffold tokens (in contrast to the regular evaluation where *randomly* 40% of the tokens were masked). In general, this task is more challenging because the molecule is more constrained. On average, less tokens are being masked and in most cases the full range of drug-likeness cannot be captured, given the scaffold. This explains the higher percentage of molecules where the primer did not influence the generations (cf. Table 5.5). Moreover, note that this includes cases where the

α	Task	0-Var (↓)	Spearman's ρ (↑)
0	Masking non-scaffold	8.55%	0.136
1	Masking non-scaffold	9.76%	0.105
0	Masking randomly	$0.80\%_{\pm 0.19}$	$0.108_{\pm 0.01}$
1	Masking randomly	$1.14\%_{\pm 0.19}$	$0.085_{\pm 0.02}$

Table 5.5: Scaffold hopping performance for SMILES model. No numerical encodings were used. No standard deviations are available for the scaffold results since the masking is deterministic.

molecule is itself a scaffold and thus no tokens are masked (we do not control for that explicitly). The generations for one exemplary molecule are shown in Figure 5.5. In this example, it is interesting to see that the model decorated the scaffold with specific atoms on the rightmost six-ring. These atoms, iodine, chlorine and bromine which were rightfully provided from low to high QED primers seem to be indicative of different levels of drug-likeness. One drawback, however, is that the RT cannot fill no or multiple tokens in the position of one `[MASK]` location. For example, in the case of the last primer (0.86), the provided scaffold already had a QED of 0.87 and thus not adding any new atoms would have been the best choice here.

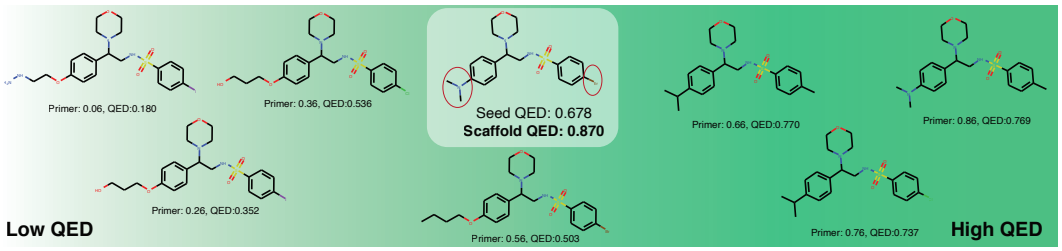


Figure 5.5: Molecules sampled in a scaffold hopping task. Only non-scaffold tokens (encircled in red) were masked.

5.3.3 Learning embeddings of numbers.

We sought to understand why the ablation studies on the numerical encodings (NE) on the QED dataset (Table 5.1 and Table 5.3) reveal only mild but not enormous superiority of models with NEs. Interestingly, we observed that in the absence of static NEs, the model learns the natural ordering of digits from the data, as shown in Figure 5.6.

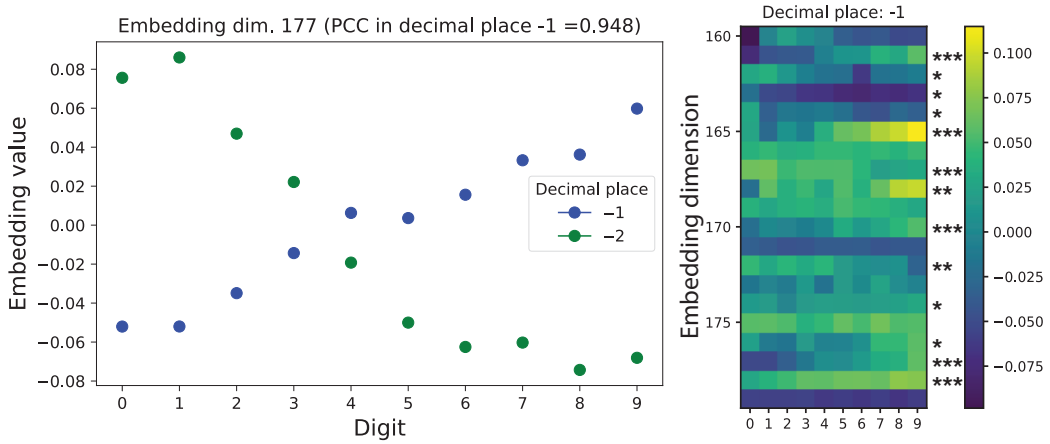


Figure 5.6: Learned embeddings of numerical tokens. *Left:* For an exemplary dimension, embeddings for 20 tokens, corresponding to 10 digits and 2 decimal places are shown. *Right:* Embeddings for 20 exemplary dimensions across 10. The stars indicate the significance level of the Pearson correlation. The analysis is based on a SELFIES model without static NEs (PLM objective).

A large number of embedding dimensions (47% and 36% for the decimal places -1 and -2 respectively) directly and significantly encoded the ordering of digits (i.e., $p < 0.05$ and $|PCC| > 0.62$ between the 10 embedding values and a strictly monotonic vector). For example, in Figure 5.6 (*left*) the digit value is monotonically related to its embedding value. Notably, this ordering trend was much less present in the models using NEs ($\sim$ 16%). For reference, with random weights, 5% would be expected.

5.3.3.1 Attention analysis

In general it is widely known that attention weights in Transformers can capture complex semantics such as protein folding structure [363] or atom-mapping in chemical reactions [29].

We therefore visualized the attention scores of the Regression Transformer using BertViz [364]. Here, we aimed to compare the inference patterns across the two tasks, property prediction and conditional generation. The results for the first 4 (out of 32) layers are shown in Figure 5.7. In general, many attention patterns commonly described

in natural language models are also present in the Regression Transformer. For example the bag-of-words pattern (i.e., evenly distributed attention, e.g., all heads of first layer) or the next-token (e.g., layer 4, head 4 and 5) or previous-token patterns (e.g., layer 2, head 2) are clearly visible. While the named patterns are consistently present in both tasks, probably because they are useful irrespective of the particular task, some distinctive patterns for either of the tasks can be found. For example, in the conditional generation task (Figure 5.7, right) many triangles with their right angle in the upper right are present. In these positions the property tokens are present and thus these patterns indicate that the representation of all other tokens, especially also the masked ones, are heavily influenced by the property value. Instead, in the property prediction task (Figure 5.7, left), many triangles with their right angle in the lower right are present. This implies a heavy attention on the `[END]` token which marks the end of the sequence and is a useful indicator for the QED score because it is critically influenced by the size/weight of the molecule. One particularly interesting attention head is head 3 in layer 2. In the property prediction task its role is to make the masked property tokens aware of the sequence length. In the conditional generation task, its role is to make all tokens aware of the property values.

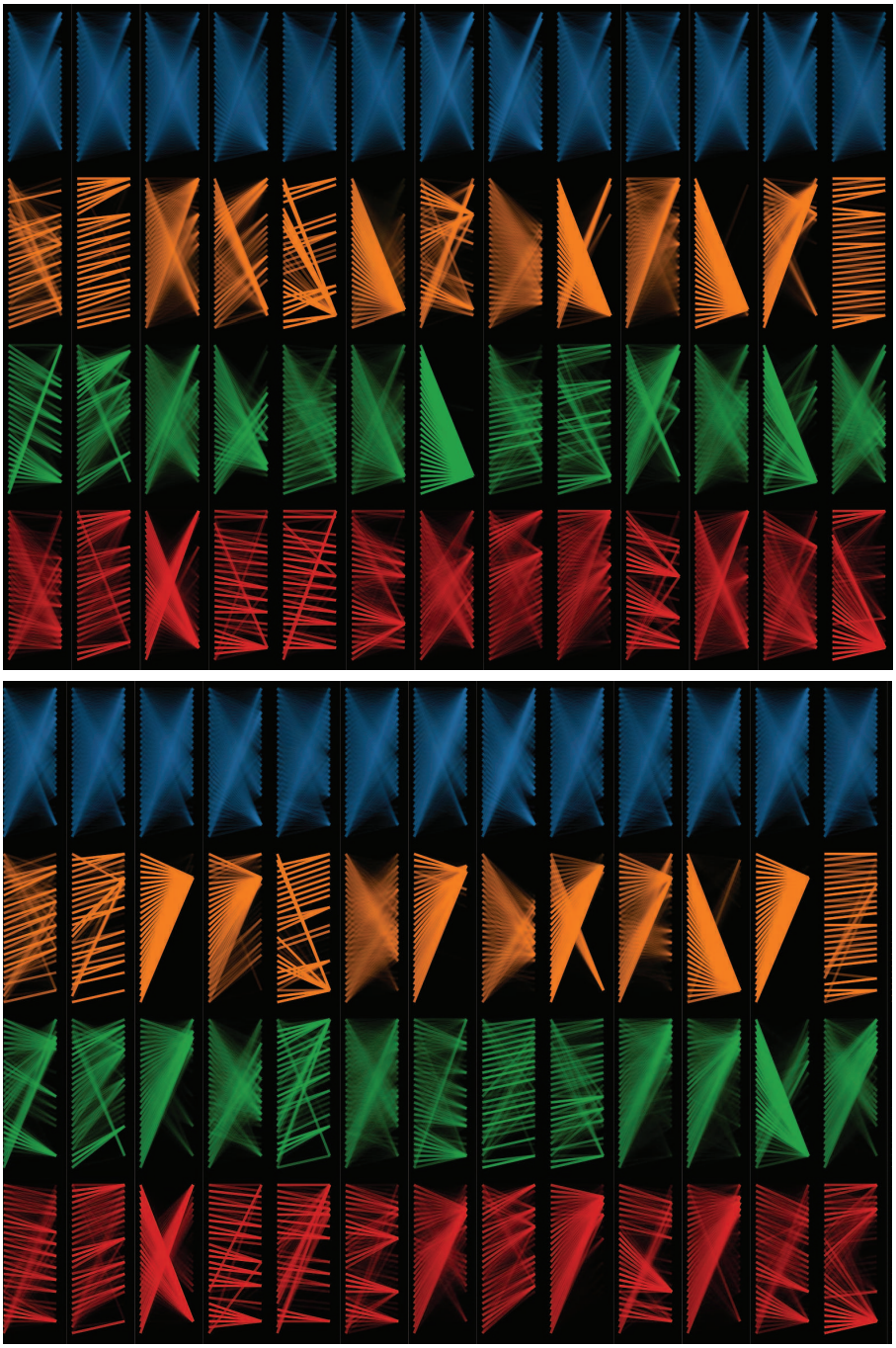


Figure 5.7: Comparing attention scores across both tasks with BertViz [364]. Attention scores for all heads of the first four layers. Rows depict layers, column depict attention heads. Within each cell, the tokens are ordered from top to bottom. *Top:* Property prediction task. *Bottom:* Conditional generation task. Plot performed with SELFIES model with float encodings, trained on the self-consistency loss.

5.3.4 Regression benchmark (MoleculeNet)

After these successful initial experiments on synthetic data, we evaluated the RT on three molecular property datasets from the MoleculeNet benchmark [34]. The regression performance on ESOL, FreeSolv and Lipophilicity is shown in Table 5.6 and compared to prior work. The strongest baseline model from MoleculeNet, XGBoost, is outperformed

Model	$\mathcal{L}_{Reg}$	ESOL	FreeSolv	Lipo.
RF [34]	✓	$1.16 \pm_{0.15}$	$2.12 \pm_{0.68}$	$0.78 \pm_{0.02}$
XGBoost [34]	✓	$1.05 \pm_{0.10}$	$1.76 \pm_{0.21}$	$0.84 \pm_{0.03}$
MPNN [34]	✓	$0.55 \pm_{0.02}$	$1.20 \pm_{0.02}$	$0.76 \pm_{0.03}$
SMILES-BERT [342]	✓	$0.47 \pm_{0.05}$	$0.81 \pm_{0.09}$	–
Mol-BERT [365]	✓	$0.53 \pm_{0.04}$	$0.95 \pm_{0.33}$	$0.56 \pm_{0.03}$
XLNet (ours)	✓	$\mathbf{0.69} \pm_{0.01}$	$\mathbf{1.03} \pm_{0.25}$	$\mathbf{0.74} \pm_{0.02}$
RT ($\alpha = 0$, NE: ✗)	✗	$0.76 \pm_{0.05}$	$1.19 \pm_{0.29}$	$0.76 \pm_{0.03}$
RT ($\alpha = 1$, NE: ✗)	✗	$0.75 \pm_{0.04}$	$1.32 \pm_{0.39}$	$0.76 \pm_{0.03}$
RT ($\alpha = 0$, NE: ✓)	✗	$0.71 \pm_{0.04}$	$1.40 \pm_{0.47}$	$\mathbf{0.74} \pm_{0.05}$
RT ($\alpha = 1$, NE: ✓)	✗	$0.73 \pm_{0.04}$	$1.34 \pm_{0.29}$	$\mathbf{0.74} \pm_{0.03}$

Table 5.6: RMSE (↓) in predicting MoleculeNet dataset properties. Performance on three different datasets across predictive models. By $\mathcal{L}_{Reg}$ we denote whether a given model used an objective function that relied on regression. All models used repeated random splits. The BERT-based models are not directly comparable to the RT, hence they are not bolded even though they performed the best. For details see text. NE means numerical encodings and α refers to the loss function in Equation 5.7.

by all our models on all tasks. Even the MPNN [85], a message-passing GNN, is slightly surpassed on FreeSolv and Lipophicility by some of our models. However, all our models are outperformed by BERT-based approaches [341, 342]. Notably, these models leveraged large-scale self-supervised pretraining before finetuning a regression head. Since these results might not be directly comparable to the RT with its XLNet backbone, we also finetuned a XLNet model with a conventional regression head. Notably, despite the absence of a regression los, the RT is on par (*Lipophilicty*) or only mildly inferior (i.e., within standard deviation range; *ESOL*, *FreeSolv*) to XLNet. But in stark contrast to all those approaches, only the RT can also be used to conditionally *generate* molecules similar to the training samples (cf. Table 5.7). Since the properties of the generated molecules are intractable to evaluate *in-silico*, we could predict them, handily, using the RT. However, as this might be a biased estimator, we evaluated them using Grover [36], a self-supervised Graph Transformer. Hence, the Spearman correlations reported in Table 5.7 are based on Grover's predictions. Overall, the generative results underline the benefit of the self-consistency loss ($\alpha = 1$) and demonstrate that the RT can adapt unseen seed molecules even according to complex molecular properties like water solubility. While we obtained

Model	NE	α	ESOL		FreeSolv		Lipophilicity	
			0-Var	ρ	0-Var	ρ	0-Var	ρ
RT	✗	0	**4.4**%	0.44	7.9%	0.53	3.6%	0.29
RT	✗	1	5.9%	0.46	7.5%	0.56	**2.7%**	**0.35**
RT	✓	0	6.1%	0.46	8.9%	**0.57**	4.2%	0.29
RT	✓	1	6.1%	**0.47**	**6.5%**	**0.57**	**2.7%**	0.34

Table 5.7: Conditional generation for MoleculeNet datasets. Average performances across three splits for training with alternating objectives. ρ refers to Spearman rank correlation and was evaluated with Grover [36]. Same legend like Table 5.6. Full table with standard deviations and self-evaluation with RT are in appendix Table A5.1.

the numerically best results on FreeSolv, we noticed that those molecules were smaller on average (15 tokens, compared to 20/47 for ESOL/Lipo). Hence, the better performance might be due to the fact that smaller molecules are easier to adapt for certain properties because changes in individual positions are more relevant. For a qualitative evaluation, we depict the generations for one exemplary seed molecule of the solubility dataset in Figure 5.4 (*bottom*). Last, corroborative for our work was the high correlation of our property predictions (RT) with Grover's for molecules generated by the ESOL, FreeSolv and Lipo models (0.86, 0.84 and 0.75 respectively). Thus, the Spearman correlations obtained with RT predictions are consistent to Grover (cf. Table A5.1).

5.4 Benchmarking against conditional generative models

To assess whether the RT can compete with conditional generative models, we benchmarked it on a property-driven molecular generation task, namely pLogP constrained optimization [19].

5.4.1 Data preparation and evaluation procedure

Data

This is a benchmark for property-driven, conditional molecular generation. The goal is to adapt a seed molecule such that a property is maximized while adhering to a fixed similarity constraint. We obtained the data from *Jin* et al. [19] which ships with a fixed split of 215,381 training and 799 test molecules and their penalized LogP (pLogP) value [366]. pLogP is the octanol-water partition coefficient (logP) penalized by the synthetic acces-

sibility score (SAS, as proposed by *Ertl* and *Schuffenhauer* [367]) and the number of cycles with > 6 atoms:

$$\text{plogp}(m) = \text{logP}(m) - \text{SA}(m) - \text{largecycles}(m) \tag{5.11}$$

Hence, pLogP just like QED can be computed directly from the molecule.

Procedure

For this task, the models were also warm-started using the QED initialization and trained for 50k steps with early stopping on perplexity. To assemble the candidates for the optimization of one seed molecule, we tried to follow the process of *Jin* et al. [19] as closely as possible. They applied 80 gradient steps, then decoded 80 molecules and reported the molecule with the highest pLogP score that satisfies the similarity constraint δ. Instead of explicit optimization, we form a pool of molecules by simply prompting 80 times with the same seed molecule but varying the fraction and the maximum span of masked tokens. From the pool of decodings we report the molecule with the highest pLogP, just like *Jin* et al. [19] and *You* et al. [25]. After this process, we report the same metrics as in their work:

1. The success rate in generating molecules with higher plogP (while adhering to the similarity constraint δ),
2. The Tanimoto similarity δ to the seed molecule,
3. The average improvement in plogP compared to the seed molecule.

5.4.2 Results

Given a seed molecule and a similarity constraint to the seed molecule (δ, given in Tanimoto similarity), the goal in this task is to generate molecules with higher pLogP values. The results in Table 5.8 demonstrate that, for both similarity thresholds δ, the RT obtained the best results. Across both similarities, it outperforms a Junction-Tree-VAE [19] and a GCPN by 614% and 103% in average improvement, respectively. While the success rate of GCPN is higher than ours, we emphasize that both JT-VAE and GCPN applied gradient optimization schemes at *inference time*. Instead, the RT does not only not require any optimization at this stage, but it was also never trained explicitly to produce molecules with high pLogP. This finding demonstrates that the RT is able to compete with specialized conditional generative models in goal-directed molecular generation. At the same time, the RT also predicted the pLogP value with a Pearson's correlation of 0.92, a task that cannot be addressed with normal conditional generative models. The results in Table 5.8 were obtained with the RT including a self-consistency loss.

In addition, we conducted experiments on lower similarity thresholds, namely $\delta = 0.2$ (cf. Table 5.10) and $\delta = 0$ (cf. Table 5.9). In these tables, we also performed an

Model	*Generation task* Improvem.	Similarity δ	Success	*Regression* PCC
JT-VAE [19]	$0.84_{\pm 1.5}$	$0.51_{\pm 0.1}$	83.6%	*Unfeasible*
GCPN [25]	$2.49_{\pm 1.3}$	$0.47_{\pm 0.1}$	**100%**	*Unfeasible*
RT (Ours)	$\mathbf{3.16}_{\pm 1.5}$	$\mathbf{0.54}_{\pm 0.1}$	97.1%	$\mathbf{0.92}_{\pm 0.0}$

(a) Similarity threshold $\delta = 0.4$

Model	*Generation task* Improvem.	Similarity δ	Success	*Regression* PCC
JT-VAE [19]	$0.21_{\pm 0.7}$	$\mathbf{0.69}_{\pm 0.0}$	46.4%	*Unfeasible*
GCPN [25]	$0.79_{\pm 0.6}$	$0.68_{\pm 0.1}$	**100%**	*Unfeasible*
RT (Ours)	$\mathbf{2.21}_{\pm 1.3}$	$\mathbf{0.69}_{\pm 0.1}$	81.8%	$\mathbf{0.92}_{\pm 0.0}$

(b) Similarity threshold $\delta = 0.6$

Table 5.8: Constrained property optimization benchmark. GCPN stands for graph-convolutional policy network [25]. JT-VAE stands for Junction-Tree Variational Autoencoder [19].

ablation study on the impact of the self-consistency loss function and the use of numerical embeddings. The results in both tables indicate that the RT consistently outperformed the JT-VAE and GCPN in the main metric (improvement) by a wide margin.

Configuration Model	NE	α	*Generation task* **Improvement**	Similarity δ	Success rate	*Regression* Pearson's r (PCC)
JT-VAE	–	–	$1.91_{\pm 2.0}$	$0.28_{\pm 0.2}$	97.5%	*Unfeasible*
GCPN	–	–	$4.20_{\pm 1.3}$	$\mathbf{0.32}_{\pm 0.1}$	**100%**	*Unfeasible*
RT	✓	1	$\mathbf{8.67}_{\pm 2.5}$	$0.10_{\pm 0.1}$	**100%**	0.92
RT	✓	0	$7.96_{\pm 2.6}$	$0.11_{\pm 0.1}$	**100%**	0.90
RT	✗	1	$8.52_{\pm 2.5}$	$0.10_{\pm 0.1}$	**100%**	0.91
RT	✗	0	$8.35_{\pm 2.6}$	$0.10_{\pm 0.1}$	**100%**	**0.94**

Table 5.9: No similarity threshold ($\delta = 0.0$).

Configuration			*Generation task*			*Regression*
Model	NE	α	**Improvement**	Similarity δ	Success rate	Pearson's r (PCC)
JT-VAE	–	–	$1.68_{\pm 1.9}$	$0.33_{\pm 0.1}$	97.1%	*Unfeasible*
GCPN	–	–	$4.12_{\pm 1.2}$	$0.34_{\pm 0.1}$	**100**%	*Unfeasible*
RT	✓	1	$\mathbf{4.45}_{\pm 1.7}$	$0.35_{\pm 0.1}$	99.6%	0.92
RT	✓	0	$4.12_{\pm 1.7}$	$\mathbf{0.36}_{\pm 0.1}$	99.6%	0.90
RT	✗	1	$4.34_{\pm 1.6}$	$0.35_{\pm 0.1}$	99.9%	0.91
RT	✗	0	$4.40_{\pm 1.7}$	$0.35_{\pm 0.1}$	99.7%	**0.94**

Table 5.10: Similarity threshold $\delta = 0.2$.

5.5 Protein language modeling application

5.5.1 Data preparation and evaluation procedure

Synthetic Boman dataset

Data. As a large-scale, labelled dataset we focused on the Boman index, a measure of potential protein interaction for peptides. It is the average of the solubility values of the residues [368]. We collected all 2,648,205 peptides with 15 to 45 AAs from UniProt [369], computed their Boman index, and used 10k and 1k samples respectively for testing and validation.

Procedure. To model protein sequences, we started with training on the Boman dataset. We trained three groups of models, one for the vanilla PLM objective (Equation 5.3) and two for the alternating objectives. We again alternated every 50 steps between optimizing Equation 5.5 and Equation 5.7 and trained one set of models with and one set without the self-consistency loss, such that $\alpha = 1$ and $\alpha = 0$ respectively in Equation 5.7. Models were trained until validation perplexity saturated ($\sim$ 4days, single GPU). The numerical values of the Boman index, originally in the range $[-3.1, 6.1]$ were normalized to $[0, 1]$ and rounded to three decimal places.

TAPE benchmark

Data. We focused on two datasets from the TAPE benchmark [333]: *Fluorescence* [370] and *Stability* [371]. The goal is to predict, respectively, the fluorescence and intrinsic folding stability of a protein that is one to four mutations away from a training protein. Both datasets ship with fixed splits. The fluorescence (stability) dataset has $21,446$ $(53,416)$ training, $5,362$ $(2,512)$ validation and $27,217$ $(12,851)$ test samples.

Procedure. For both datasets, three models were warm-started using the Boman initialization and trained until validation performance saturated ($\sim$ 100k steps). We

used the self-consistency objective for all experiments. The numerical values were again scaled to $[0, 1]$. On the Fluorescence data, a small value of Gaussian noise was added to some training samples due to an interesting failure mode (discussed later in Subsubsection 5.5.3.1). For the evaluation of the conditional generation task, the models were given more flexibility: 60% of the tokens were masked (i.e., $c = 1.7$ in Equation 5.3) and the maximum span was 7 amino acid residues. We did not evaluate the RT on conditional generation for the Fluorescence dataset because of a massive pretraining-finetuning mismatch: While the Boman dataset used for pretraining consisted of 15 to 45 residues (mean/std: 36 ± 7), the fluorescence proteins were significantly larger (246 ± 0.2 residues). Instead, the proteins in the stability dataset were similar in size to the pretraining data (45 ± 3 residues).

5.5.2 Synthetic pretraining: Boman index

To assess the generality of the RT beyond chemical languages, we benchmarked the RT in protein language modeling. On the synthetic pretraining data, the RT obtained nearly perfect results in predicting Boman's index (Spearman $\rho > 0.994$; Table 5.11) and

		Regression task		*Generation task*	
Model	Loss	RMSE (↓)	Pearson's r (↑)	0-Var (↓)	Spearman ρ (↑)
k-NN	–	0.53	0.932	*Task unfeasible*	
RT	PLM	$0.69_{\pm 0.03}$	$0.944_{\pm 0.0}$	$0.3_{\pm 0.4}$	$0.76_{\pm 0.03}$
RT	$\alpha = 0$	$\mathbf{0.17}_{\pm 0.04}$	$\mathbf{0.994}_{\pm 0.0}$	$\mathbf{0.2}_{\pm 0.1}$	$0.82_{\pm 0.01}$
RT	$\alpha = 1$	$0.20_{\pm 0.04}$	$0.991_{\pm 0.0}$	$\mathbf{0.2}_{\pm 0.1}$	$\mathbf{0.84}_{\pm 0.00}$

Table 5.11: Ablation study on training schemes for Boman dataset. Again, α refers to the self-consistency objective in Equation 5.7.

outperformed a baseline k-NN using Levenshtein distance [161]. This is a meaningful baseline model because the Boman index solely depends on the frequencies of amino acids. These results confirmed that the superiority of the self-consistency objective also extends beyond the domain of small molecule modeling. Like on the QED dataset, the self-consistency loss led to better results in conditional generation, but at the expense of slightly reduced accuracy in regression. We believe that this might be caused by the self-evaluations of the decoded sequences. These sequences might differ significantly from the training sequences but are still used with the property value of the original sequences.

Moreover, the RT also successfully generated peptides with a desired Boman index given a partially corrupted amino-acid sequence (cf. Spearman ρ of 0.84, see Table 5.11). Apart from that, Figure 5.8 reveals a general trend in the conditional generation with the Regression Transformer: More freedom in the generative process (i.e., a higher fraction of masked amino acid residues) leads to better results in terms of Spearman ρ to the property

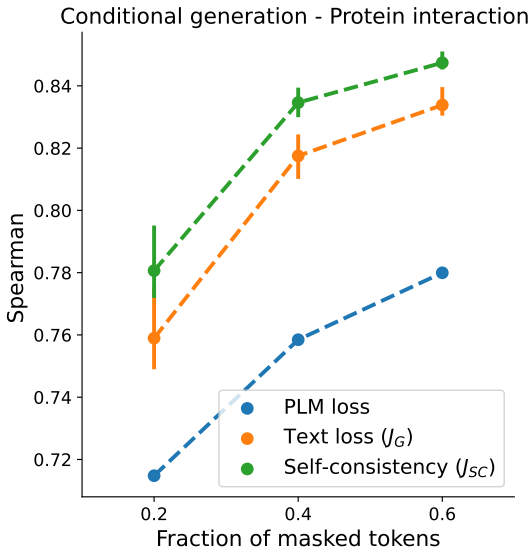


Figure 5.8: Correlation between property primer and property of generated protein sequences The model's ability to generate protein sequences with a desired protein interaction index. The self-consistency loss yielded the best results and, generally, a higher fraction of masked tokens led to generated peptides that adhere better to the primed property value.

primers (cf. Figure 5.8). This comes, however, at the cost of reduced similarity to the seed sequence.

5.5.3 Protein fluorescence and stability

Next, we evaluated the RT on protein property prediction benchmarks from TAPE [333]. As can be seen in Table 5.12, the RT performed competitively on these two realistic protein regression datasets. This is remarkable given that the TAPE models were pretrained

Model	Source	Fluoresc.	Stability
k-NN	Baseline	0.59	0.21
One-Hot	TAPE	0.14	0.19
LSTM	TAPE	0.67	0.69
Transformer	TAPE	0.68	**0.73**
UniRep	[372]	0.67	**0.73**
RT	**Ours**	$\mathbf{0.72}_{\pm 0.04}$	$0.71_{\pm 0.02}$

Table 5.12: Protein regression tasks. All values in Spearman's ρ (↑) on the test set. TAPE datasets/performances taken from *Rao* et al. [333].

large-scale on unlabelled protein sequences and finetuned with a regression loss. For ex-

ample, the RT outperforms all reported methods in Spearman correlation on the Fluorescence task. The competitive predictive performance of the RT demonstrates that the benefits of self-supervised pretraining can extend to numerically labelled datasets. Instead of self-supervised pretraining on an *unlabelled* dataset of protein sequences, like commonly done in prior art [333, 372, 373], the RT can perform self-supervised pretraining on a numerically labelled dataset. This large-scale pretraining on the Boman dataset yields, *en passant*, a conditional generative model for property-driven local exploration of the protein sequence space. Evidence on the peptide generation results based on the stability dataset can be found in Table 5.13. Whereas all TAPE models as well as the UniRep method

Model	Stability dataset	
	0-Var (↓)	Spearm. ρ
All TAPE	*Task unfeasible*	
UniRep		
RT	$19\%_{\pm 4.5}$	$0.44_{\pm 0.01}$

Table 5.13: Protein generation performance. Standard deviations measured across three runs.

are incapable of addressing this generation task, the RT was able to modify the test peptides such that their (predicted) stability correlated strongly with the primed property ($\rho = 0.44$).

5.5.3.1 Can we truly "regress"?

Arguably the most interesting aspect about the RT model is the unconventional way of solving a regression task by predicting a sequence of tokens (i.e., characters) that (hopefully) translate to a valid floating-point number. Given that the RT is trained on a conventional cross-entropy loss which cannot convey similarity between tokens, it is clear that the RT will always suffer from a training-evaluation-mismatch. The RT is trained as a classification model but during evaluation, it is assessed with regression metrics such as RMSE or Spearman's ρ. We therefore sought to understand to what extent the RT can truly "regress".

To examine that, the protein fluorescence dataset from *Sarkisyan* et al. [370] is particularly interesting because it has a bimodal mode: one mode corresponding to bright proteins, the other to dark proteins. During initial training, we observed an interesting failure mode. Figure 5.9 shows that the dark mode has one sharp spike, exactly at a log fluorescence value of 1.301. Almost 10% of all training samples and almost 50% of the proteins in the dark mode have this *exact* value. The Regression Transformer is trained on a classification loss and so, the loss during training for such samples will be distributed across the five tokens `1_0`, `_.`, `3_0`, `0_-1` and `1_-2`. In many cases, the model collapsed to always predicting 3.301 where the first token (`3_1`) was correct for all samples in the bright mode and the remaining tokens (`3_0`, `0_-1` and `1_-2` were correct for most samples in the dark

model. As a non-algorithmic remedy, we added Gaussian noise to those training samples as described above. Note that we do not apply any weighting of the individual numerical tokens.

Upon this perturbation, the training performance converged more smoothly. When inspecting the predictions of the well-behaving models in more depth, we found that the RT excels at recognizing the mode of a protein but struggles with intra-mode precision. This can be seen in Figure 5.9 which reveals the improved performance of the RT compared to the finetuned TAPE Transformer: Less samples were predicted in the wrong mode. However, the RT had difficulties with a fine grained regression, in particular in

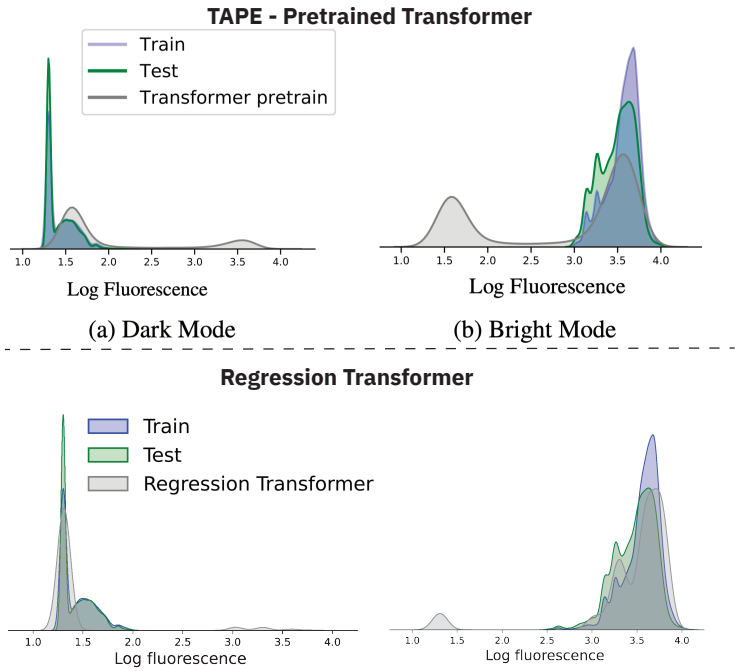


Figure 5.9: Bimodal mode of fluorescence data. The upper part of the plot has been copied from *Rao* et al. [333] (Figure 3). It shows the bimodal mode of the training data and the test predictions from the TAPE Transformer. At the bottom, we show our remake of the above plot by replacing the predictions from the pretrained TAPE Transformer with the predictions from the Regression Transformer.

the bright mode. This becomes particularly apparent when inspecting the detailed results, grouped by bright and dark test proteins respectively in Table 5.14. While the RT achieved the best results in the overall Spearman ρ, the recommended metric by *Rao* et al. [333], it does not dominate any of the mode-specific metrics. This is a noteworthy finding because it reflects the tendency of the RT to strive for a multi-class classification rather than performing a full regression. It is also interesting to see that the baseline models (k-NN and TAPE One-Hot) achieved the best results in MSE of bright proteins.

		Full test set		Bright proteins		Dark proteins	
Model	Source	MSE	ρ	MSE	ρ	MSE	ρ
One-Hot	TAPE	2.69	0.14	0.08	0.03	3.95	0.00
k-NN	**Ours**	2.31	0.59	**0.05**	0.30	3.37	0.04
Pretr. LSTM	TAPE	**0.19**	0.67	0.12	0.62	**0.22**	0.04
Pretr. Transf.	TAPE	0.22	0.68	0.09	0.60	0.29	**0.05**
UniRep	UniRep	0.20	0.67	0.13	**0.63**	0.24	0.04
RT	**Ours**	0.34	**0.72**	0.19	0.45	0.40	0.04

Table 5.14: Detailed fluorescence prediction results. MSE abbreviates mean squared error. TAPE and UniRep performances taken from *Rao* et al. [333]. For the RT all standard deviations on ρ and MSE were < 0.05 and < 0.1 respectively.

5.6 Chemical reaction modeling applications

The two tasks that we performed with the RT on chemical reactions are visualized in Figure 5.1**f**. The vanilla task to predict the yield of a reaction from its reaction SMILES. The alternative, generative task was to generate a novel precursor, based on the rest of the reaction (i.e., the remaining precursors, the product and the yield).

5.6.1 Data preparation and evaluation procedure

Pretraining on USPTO reactions

Before training on the narrow yield datasets, we warmed up the model to learn generic reaction chemistry. This was done because the two reaction yield datasets only cover narrow regions of the chemical space (one template applied to many precursor combinations).

Data. We used reactions from the US Patent Office (USPTO), the largest open-source dataset about chemical reactions [374]. Since no yield information was available, the utilized numerical property was the total molecular weight of all precursors. The dataset contained $n = 2,830,616$ reactions and was obtained from *Schwaller* et al. [29].

Procedure. 5000 reactions were held out for validation and the model was trained until validation performance on the two alternating objectives (with self-consistency loss) saturated. The masking hyperparameter c was set to 2.5 and the model were trained for $\sim$ 2 days (single GPU). The vocabulary for reaction SELFIES contained 861 tokens.

Reacion yield datasets

We investigated two high-throughput experimentation (HTE) yield datasets that examine specific reaction types: Buchwald-Hartig aminations [375] and Suzuki-Miyaura cross-

coupling reactions [376]. Both datasets were investigated in the same 10 random splits as examined in *Schwaller* et al. [62] with a 70/30% train/validation ratio.

Buchwald-Hartwig data. This dataset, produced by *Ahneman* et al. [375], investigates HTE of Palladium-catalysed Buchwald-Hartwig C-N cross coupling reactions. The reaction space comprises 3955 reactions, spanned by 15 unique aryl and heteroaryl halides, 4 Buchwald ligands, 3 bases and 22 isoxazole additives. A Palladium-catalyst and a Methylaniline are the fifth and sixth precursor respectively, however they are identical for all reactions. Each reaction is associated to a yield $y \in [0, 100]$ and the 10 random split were identical to the ones released by *Sandfort* et al. [377] that are also used by all competing methods in Table 5.15.

Suzuki cross-coupling data. This dataset was provided by *Perera* et al. [376] and investigates HTE of Suzuki-Miyaura reactions across 15 pairs of electrophiles and nucleophiles, leading to different products respectively. For each pair, a combination of 4 solvents, 12 ligands and 8 bases (reagents) was measured, resulting in a total of 5760 reaction yields that we scale to the range $[0, 100]$. The catalyst is identical for all reactions, some reactions omitted the ligand or the base while others contained electrophiles, nucleophiles, ligands, bases or solvents that were composed of different fragments (e.g., salts).

Procedure. For both the Buchwald-Hartwig reactions [375] and the Suzuki-couplings [376], ten models were finetuned respectively on repeated random splits. The training objectives again alternated every 50 steps between property prediction and conditional generation with $\alpha = 1$ for a maximum of $50k$ steps ($\sim$ 1 day). Notably, during the conditional generation task we sampled one precursor per batch and then entirely but exclusively masked this precursor. Thus the objective for the model became to reconstruct a missing precursor from the remaining precursors and the reaction yield (or to produce an alternative precursor with a similar predicted yield). We used SELFIES.

Evaluation reaction generation. In this section, we challenge the model with two sequence generation tasks.

1. Fully *reconstructing* a precursor solely based on the remaining precursors and the reaction yield. The top-3 predicted sequences (decoded via beam search) are considered, s.t. **Top-3 accuracy** is reported. Additionally we report the average **Tanimoto similarity** of the most similar of the top-3 molecules to the seed molecule (fingerprint: ECFP4).
2. Secondly, we measure the capability of *decorating* existing reactions to obtain a (potentially) higher yield. To that end, the model is prompted with incomplete reactions consisting of an increased yield, an entirely masked precursor and complete remaining

precursors. We consider the top-3 predicted sequences (decoded via beam search) and report the fraction of samples where one of the reactions had a higher (predicted) yield (**success rate**). The second response metric is the **mean improvement** in (predicted) reaction yield (yield $y \in [0, 100]$, the distributions are right-skewed). Note that we exclude trivial solutions by removing all predicted precursors that exist in the training dataset.

5.6.2 Results

5.6.2.1 Reaction yield prediction

Language models have significantly advanced reaction chemistry [29, 53] and also showed superior performance on yield prediction [62], yet models incorporating yield into (partial) reaction generation are lacking entirely. We therefore optimized the RT for concurrent yield prediction and precursor generation on two reaction-yield datasets: Buchwald-Hartig aminations [375] and Suzuki-Miyaura cross-couplings [376]. On yield prediction, the RT outperforms fingerprint-based or quantum-mechanics methods as can be seen in Table 5.15. Moreover, it matches (Suzuki dataset) or almost matches (Buchwald dataset) the performance of language models like Yield-BERT, trained with regression loss on SMILES.

Model	Buchwald-Hartwig	Suzuki-Coupling
One-Hot [377]	0.89	–
DFT [375]	0.92	–
MFF [377]	$0.927_{\pm 0.01}$	–
Yield-BERT [62]	$\mathbf{0.951}_{\pm 0.01}$	$0.79_{\pm 0.02}$
Yield-BERT finetuned	$\mathbf{0.951}_{\pm 0.01}$	$\mathbf{0.81}_{\pm 0.01}$
RT (**ours**)	$0.939_{\pm 0.01}$	$\mathbf{0.81}_{\pm 0.02}$

Table 5.15: Reaction yield prediction performance. Evaluated on ten 70/30 splits, measured in coefficient of determination (R^2).

5.6.2.2 Reconstructing precursors

The same model that was trained to predict yield in Buchwald-Hartwig animations is also able to reconstruct missing precursors, as shown in Table 5.16. This can be useful to infer missing solvents or reagents in automatically extracted reactions. We measure reconstruction performance by showing the percentage of cases where the exact right precursor was among the top-3 predicted sequences and the Tanimoto similarity of the most similar of those molecules. The reconstruction is partly achieved with great accuracy (e.g., 98.2% for aryl-halides). Interestingly, inferring additives proved challenging, the top-3 accuracy

Precursor	Top-3 accuracy	Tanimoto similarity
Halide	$98.23\%_{\pm 0.5}$	$0.991_{\pm 0.00}$
Ligand	$50.38\%_{\pm 1.6}$	$0.677_{\pm 0.01}$
Base	$100\%_{\pm 0.0}$	$1.000_{\pm 0.00}$
Additive	$1.36\%_{\pm 0.5}$	$0.158_{\pm 0.02}$

Table 5.16: Reconstructing precursors for Buchwald-Hartwig aminations [375]. Each reaction in the dataset also contained 4-Methylaniline and the same Palladium-catalyst, thus they are excluded from the analysis. Full precursors were generated.

in the reconstruction performance was $< 1\%$. We hypothesize that this might be because additives are the dominant precursor type for the reaction yield [375]. Furthermore, we observed that when we masked the additive only partially (rather than completely), the reconstruction performance increases significantly. This can be seen in the ablation study in Table 5.17.

p_{mask}	Top-3 accuracy	Tanimoto similarity
1.0	$1.36\%_{\pm 0.5}$	$0.158_{\pm 0.002}$
0.5	$11.47\%_{\pm 1.0}$	$0.316_{\pm 0.002}$
0.25	$46.74\%_{\pm 3.5}$	$0.645_{\pm 0.003}$

Table 5.17: Ablation study on model flexibility for generation of additives. Performance in generating additives for Buchwald-Hartwig reactions [375] as a function of p_{mask}, i.e., the fraction of tokens in the additive that are masked. Generation was primed with remaining precursors and yield.

On the Suzuki-couplings, the reconstruction results are more balanced among the five precursor types; the average Tanimoto similarity to the true precursor was > 0.65 in all cases (cf. Table 5.18).

Precursor	Top-3 accuracy	Tanimoto similarity
Electrophile	$44.2\%_{\pm 17.6}$	$0.732_{\pm 0.02}$
Nucleophile	$100.0\%_{\pm 0.0}$	$1.000_{\pm 0.00}$
Ligand	$67.4\%_{\pm 20.0}$	$0.689_{\pm 0.15}$
Base	$90.5\%_{\pm 1.2}$	$0.811_{\pm 0.01}$
Solvent	$56.4\%_{\pm 1.1}$	$0.661_{\pm 0.01}$

Table 5.18: Reconstructing precursors for Suzuki couplings [376]. Each reaction in the dataset also contained the same Palladium-catalyst which is thus excluded from this analysis.

5.6.2.3 Improving reactions by generating new precursors

In addition to yield prediction and precursor reconstruction, the RT can also *decorate* existing reactions by adapting specific precursors toward a higher yield. The performance on this in the Buchwald-Hartwig dataset can be found in Table 5.19. We measured decora-

Precursor	Success rate	Mean improvement
Halide	$42.3\%_{\pm 2.4}$	$6.1_{\pm 1.3}$
Ligand	$74.4\%_{\pm 4.2}$	$14.4_{\pm 1.7}$
Base	$82.2\%_{\pm 2.3}$	$8.1_{\pm 0.6}$
Additive	$71.2\%_{\pm 1.8}$	$11.7_{\pm 1.3}$

Table 5.19: Generating precursors for Buchwald-Hartwig aminations [375]. Each reaction in the dataset also contained 4-Methylaniline and the same Palladium-catalyst, thus they are excluded from the analysis. Full precursors were generated ($p_{mask} = 1$).

tion performance by reporting the percentage of cases where the top-5 predicted reactions contained a reactions with higher (predicted) yield than the seed reaction (success rate). The second reported metric is the associated average yield improvement, again given on a scale $[0, 100]$. Consistently across all precursor types, 40-80% of the top-5 predicted sequences contained reactions with entirely novel precursors and higher predicted yield. Figure 5.10 visualizes exemplary adaptations of base and arly-halide of a BH amination with very low yield ($< 5\%$). This reaction was unseen during training and had a very low yield (5%). The RT found novel adaptations of each of the four precursor types that resulted in an increase of predicted yield to 11-85% yield. With the forward reaction prediction model in IBM RXN [28] we confirmed that all reactions indeed result in the desired product. Notably, the confidence from the forward model rank-correlated almost perfectly with the yield predicted by the RT ($\rho = 0.90, p < 0.05$).

The results on the precursor generation for the Suzuki couplings are shown in Table 5.20. Similarly to the BH aminations, 50-60% of the top-5 predicted sequences contained reactions with entirely novel precursors and higher predicted yield.

Precursor	Success rate	Mean improvement
Electrophile	$63.5\%_{\pm 7.1}$	$12.5_{\pm 3.4}$
Nucleophile	$54.0\%_{\pm 6.2}$	$5.4_{\pm 0.8}$
Ligand	$56.7\%_{\pm 3.5}$	$5.5_{\pm 0.6}$
Base	$47.8\%_{\pm 2.7}$	$4.6_{\pm 0.3}$
Solvent	$57.8\%_{\pm 1.8}$	$7.5_{\pm 0.3}$

Table 5.20: Generating precursors for Suzuki couplings [376]. Each reaction in the dataset also contained the same Palladium-catalyst which is thus excluded from this analysis.

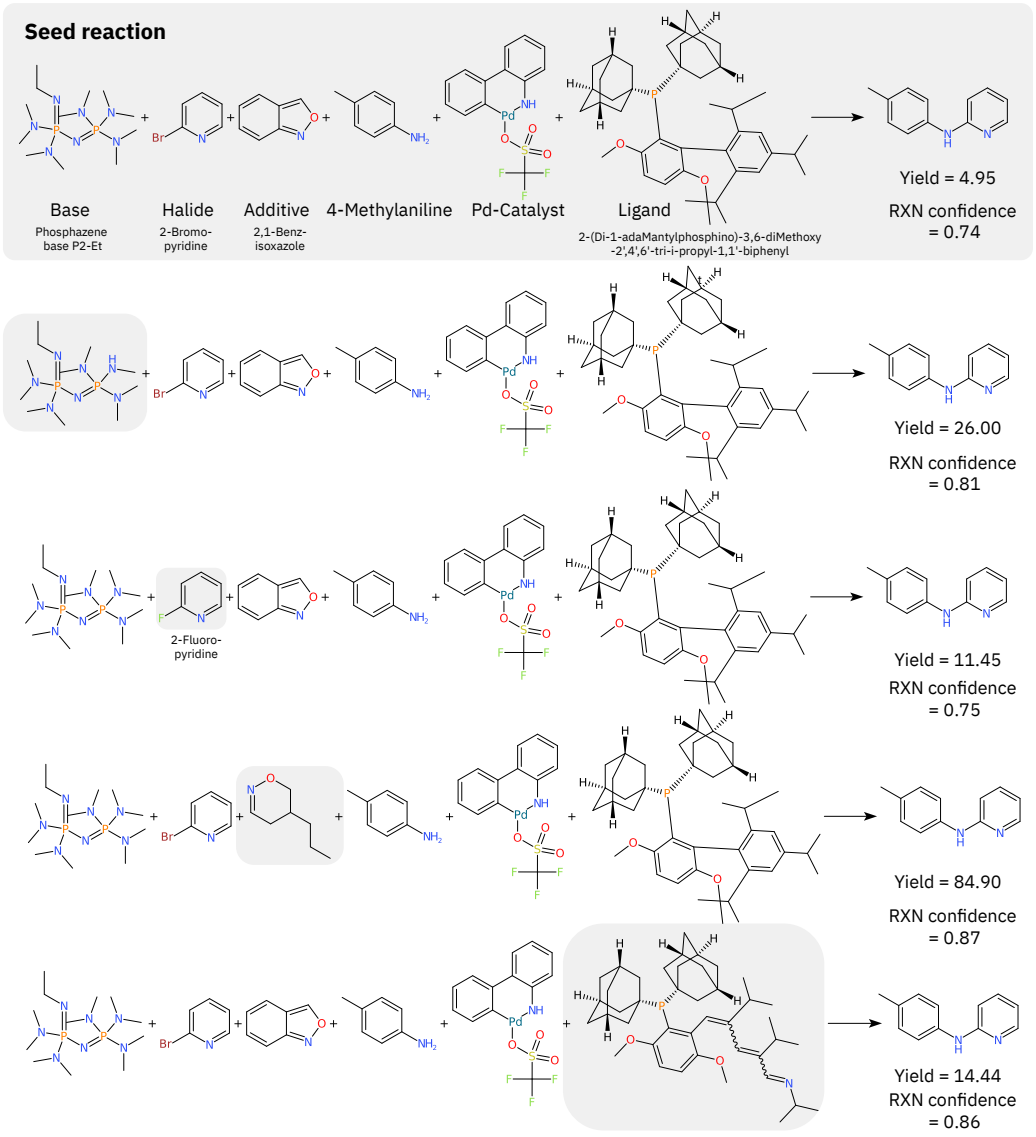


Figure 5.10: Adapting an unseen Buchwald-Hartwig amination toward higher yield. Together with a BH amination from the validation dataset (*top*), we show four RT-generated reactions with adaptations of the base, the halide, the additive and the ligand respectively. In this case, the predicted yield of all new reactions was higher. The RXN confidence stems from the forward model by *Schwaller* et al. [28] which confirmed that the reaction would result in the shown product in all cases. Note that no adaptations of 4-Methylaniline and the Palladium-catalyst can be generated since they are constant cross the dataset.

5.6.2.4 Benefit of co-encoding numerical properties

A key advantage of the RT over conventional language models is the ability to co-encode text with numerical properties. Conventional SMILES translation models like the Molecular Transformer [28] could also be used to reconstruct missing precursors but they would be unable to condition this reconstruction meaningfully on the yield.

Therefore, we conducted a last ablation study to assess the benefit of co-encoding numerical properties. The results are shown in Table 5.21 and Table 5.22 for the BH aminations and the Suzuki couplings respectively. The results consistently show mild benefits

Precursor type	Top-3 accuracy		Tanimoto similarity		Unique n
	Prec. + Yield	Precursors	Prec. + Yield	Precursors	
Aryl halide	$98.23\%_{\pm 0.5}$	$98.21\%_{\pm 0.4}$	$0.991_{\pm 0.003}$	$0.991_{\pm 0.002}$	15
Ligand	$50.38\%_{\pm 1.6}$	$50.43\%_{\pm 1.7}$	$0.677_{\pm 0.010}$	$0.678_{\pm 0.010}$	4
Base	$100.0\%_{\pm 0.0}$	$100.0\%_{\pm 0.6}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	3
Additive	$1.36\%_{\pm 0.5}$	$1.25\%_{\pm 0.8}$	$0.158_{\pm 0.018}$	$0.158_{\pm 0.019}$	22

Table 5.21: Generating precursors for Buchwald-Hartwig reactions [375] based on remaining precursors or remaining precursors *and* yield. Full precursors were generated ($p_{mask} = 1$). Unique n denotes the number of unique samples per entity in the training dataset.

Precursor type	Top-3 accuracy		Tanimoto similarity		Unique n
	Prec. + Yield	Precursors	Prec. + Yield	Precursors	
Electrophile	$44.19\%_{\pm 17.6}$	$31.39\%_{\pm 15.3}$	$0.732_{\pm 0.160}$	$0.591_{\pm 0.141}$	7
Nucleophile	$100.0\%_{\pm 0.0}$	$100.0\%_{\pm 0.0}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	4
Ligand	$67.43\%_{\pm 20.0}$	$67.59\%_{\pm 19.8}$	$0.689_{\pm 0.152}$	$0.690_{\pm 0.152}$	5
Base	$90.53\%_{\pm 1.2}$	$90.50\%_{\pm 1.4}$	$0.811_{\pm 0.006}$	$0.811_{\pm 0.001}$	8
Solvent	$56.74\%_{\pm 1.1}$	$56.52\%_{\pm 1.0}$	$0.661_{\pm 0.009}$	$0.660_{\pm 0.007}$	4

Table 5.22: Generating precursors for Suzuki-cross-couplings reactions [376] based on remaining precursors or remaining precursors *and* yield. Legend like Table 5.21.

in reconstruction performance when providing the true yield rather than masking it. This highlights the benefit of jointly encoding input and target variables.

5.7 Discussion

In the last chapter of this thesis we presented the Regression Transformer (RT). We demonstrated that regression can be casted as conditional sequence learning task and introduced a flexible multitask-language-model with wide application in scientific discovery. We hope to have shown that the RT can act as a "swiss army knife" transformer

that bridges previously considered disjoint tasks such as property prediction and conditional sequence generation. The RT was shown to excel at both tasks and could thus pave the road toward foundation models in material design.

Regarding molecular property prediction, we find that the RT learns continuous properties even from small datasets, surpasses conventional regression models on several benchmarks and sometimes competes with Transformers trained on regression loss. Remarkably, this is achieved without providing ratio-scale information about the property, potentially even challenging the necessity of using regression rather than classification objectives.

The experiments on conditional text generation underline the versatility of the RT: Across a wide range of tasks, we conditionally generated novel sequences (molecules, proteins, reactions) that seemingly adhere to primed, continuous properties. Our experiments on constrained molecular generation benchmark further demonstrate that the RT can surpass specialized conditional generative models. We foresee this to be useful for property-driven, sub-structure constrained molecular or protein design. Moreover, the RT presents a viable solution to extend self-supervised pretraining from unlabelled datasets to numerically labelled datasets. We have further demonstrated that co-encoding task-related continuous properties (such as reaction yield) can be beneficial to boost model performance in sequence generation tasks (e.g., reconstruct missing precursors). We also emphasize that even though all experiments reported herein examined singular properties, the RT naturally scales to multiproperty prediction.

Future work could, for example, intensify the work on reaction modeling (the RT effectively generalizes forward reaction and retrosynthesis models) or improve the ability of the RT to perform fine-grained regression. Finally, our work resonates with the recent trend towards multitask Transformers [331, 378, 379] and we envision it as a mean to accelerate the development of foundation models for scientific discovery applications.

6 Concluding remarks

Reflection on contributions

In this thesis, we have developed and studied generative language models for *de novo* molecular design. We strived to develop generative models that can be conditioned on a wide range of complex starting conditions (e.g., protein targets, molecular substructures or a property value with floating-point precision) and generate molecules that adhere to the starting criterion.

Before focusing on these tasks we have devoted the first part of this thesis to the development of language models to predict molecular properties such as toxicity, solubility or binding affinity to a protein. This was done in order to have more interesting and meaningful evaluation metrics than commonly employed for molecular generative models.

Interlude on evaluation metrics of generative models

> Notably, an appropriate evaluation of molecular generative models still remains difficult and has been identified as a key challenge for the field [337]. The current state of the field has aptly been summarized as:
>
> *"The current evaluations for generative models do not reflect the complexity of real discovery problems."* *Coley* et al. [380]
>
> A physical validation is entirely impractical, an analytical evaluation is impossible since the learned distributions cannot be formulated analytically. Critically, not even a (trained) human eye can act as a reliable oracle (unlike for text or image generation methods). Therefore, common evaluation metrics are extremely simplistic and include novelty, uniqueness, validity, diversity [381] or the Frechet Chemnet distance [382], basically a Wasserstein-2 distance on the activations on the penultimate layer of a neural network. *Renz* et al. [383] convincingly demonstrated that all those metrics are practically useless as a simple algorithm that randomly inserts carbon atoms to molecules from the training data outperforms models previously considered as SOTA. Throughout this thesis we have thus refrained from assessing our models by such metrics and rather attempted to build complex molecular property predictors that can serve as evaluation functions. In the field, established

metrics for biochemical metrics are still largely absent. Performance on optimization of simple chemo-centric properties like QED or logP are frequently reported even though their practical use is marginal. For some subdomains of molecular discovery somewhat accepted evaluation metrics exist, e.g., molecular docking for protein-targeted drug discovery. However, such evaluations are computationally extremely demanding and thus not applicable large-scale. Recently, *Cieplinski* et al. [384] found that most SOTA methods did not succeed in generating molecules with high docking scores and proposed a benchmark for faster docking score evaluation. In our recent work by *Tadesse*, *Born*, et al. [385], we therefore proposed MPEGO, a first step toward a hierarchical and flexible evaluation of generative models. The main advantage regarding the flexibility is that users can select and weight properties to be considered for evaluation. Upon a selection of one or multiple properties (they can be everything from molecular weight to the user-provided outcome of a real experiment, all we require is to have the property available for a set of training molecules as well as generated molecules), MPEGO compares the property distributions in a hierarchical manner. This hierarchical evaluation can be done either based on a feature discretization scheme or by directly comparing the distributions with the Wasserstein distance.

The first part of this thesis has shown that competitive performance in molecular property prediction can be achieved with language models that solely rely on SMILES and protein sequences. We benchmarked various molecular representations (e.g., fingerprints, different flavors of SMILES and SELFIES, as well as graph and graph kernel methods) and revealed that SMILES coupled with augmentation overall obtained the best performance. Moreover, the attention weights of our model, ToxSmi, allowed for easy interpretation and showed enrichment of known toxicophores even without explicit supervision. We introduced a notion of model reliability by proposing and combining two simple methods for uncertainty estimation (Monte-Carlo dropout and test-time-augmentation) and found that those methods not only identify samples with high prediction uncertainty, but also allow forming implicit model ensembles that improve accuracy. We validated ToxSmi on a large-scale proprietary toxicity dataset and find that it outperforms previous work while giving similar insights into revealing cytotoxic substructures.

The next objective was to investigate the task of protein-ligand binding affinity prediction and develop a multimodal proteochemometric language model. We challenged a common practice in sequence-based CPI prediction models, i.e., relying on full protein sequences. By representing kinases only through 29 residues comprising the ATP binding site, we disregarded $> 95\%$ of the commonly considered protein features and found that this significantly and robustly improves model performance. This is probably due to an increased SNR as well as an implict incorporation of 3D information into the 1D model stemming from the discontiguity of the active site in the original sequence. We proposed several new sequence augmentation strategies that yield

complementary performance benefits.

In the second part of this thesis, we first developed a hybrid VAE that bridges systems biology and molecular design. By fusing the latent spaces of two separately pretrained VAEs this model can effectively incorporate system-level information about the target environment of the a molecule into the generative process. This method was first exemplified for the discovery of anticancer hit molecules and steered the molecular generation by exploiting the previously proposed toxicity and drug sensitivity prediction models as reward functions. In our experiments, the molecule generation could be biased to molecules with high predicted drug sensitivity for specific cell line profiles or cancer types. The molecules that were generated in a cancer-type-specific-manner were further analyzed and often showed the highest structural similarity to existing drugs with known relations for exactly those cancer types. Even though those results were promising, we emphasize that without a successful wet-lab validation the true value of any molecular generative model remains unclear. Within our proposed framework, the quality of the generative model is inherently constrained by the predictive power of the reward function and thus satisfying extrapolation capabilities of the reward functions are instrumental for this hybrid VAE.

We then extended this framework to protein target-driven molecular design and apply it on 41 SARS-CoV-2 related proteins. The results demonstrated high generalization capabilities and showed that our method does not require finetuning for specific targets. In a leave-one-out cross validation we found that molecules with high predicted binding affinity were proposed even against unseen protein targets. The proposed molecules showed a comparably high selectivity while being more promiscuous to other SARS-CoV-2 targets than other molecules. For the first time we reported the coupling of 1) a molecular generative model, 2) tools for automated synthesis planning and 3) the successful synthesis on a robotic hardware. Even though we lacked the experimental validation, this was a modest step toward accelerated molecular discovery through a completely autonomous workflow that did not require human intervention from *hypothesis* to the *make*.

In the last chapter of this thesis we proposed the *Regression Transformer*, a multitask language model that seamlessly bridges sequence regression (i.e., property prediction) and conditional sequence generation. We thoroughly demonstrated that, despited using a nominal-scale training objective, the Regression Transformer matched or surpassed the performance of conventional regression models in property prediction trasks of small molecules, proteins and chemical reactions.

Our main motivation for this method, however, was to incorporate an inductive bias about continuous properties of interests into a generative language model. Upon priming our model with floating-point property constraints, we obtained a competitive con-

ditional generative model that outperformed specialized approaches in a substructure-constrained property-driven optimization benchmark. This dichotomy was facilitated by a novel training scheme that optimizes sequence regression and generation in an alternating fashion. Across the domains of small molecules, proteins and chemical reactions we found that the RT can decorate arbitrarily corrupted seed sequences by desired properties. The RT thus constitutes a highly flexible model that will hopefully find wide application for exploring the local chemical space around a seed molecule in a property-constraint manner.

Closing the loop

While the methods proposed in this thesis can intertwine molecular discovery tasks better with virtual screening methods, the molecular discovery loop (cf. Figure 1.1**C**) yet has to be closed. This can be achieved with traditional techniques such as manual planning and execution of synthesis and *in vitro* experimentation. However, synergies will certainly be higher if the remaining parts in the DMTA cycle were also autonomous. This might be cheaper and faster and would certainly yield more reproducible and standardized experimental results. *Gromski* et al. [386] envisioned a tight integration of generative algorithms into laboratory workflows that allowed refinements based on synthesis and experimental results in closed-loop, autonomous manner.

Traditionally, computer-aided synthesis planning (CASP) largely relied on expert-crafted reaction rules [387]. Recently however, significant progress has been achieved in automated synthesis planning [28, 30, 60, 388]. Lots of this progress was due to the transfer of methods from NLP to the field of chemistry. By treating chemistry as a natural language, the Molecular Transformer [28] enabled a fully autonomous planning of multistep synthesis routes without any injection of expert knowledge [60]. However, even upon the identification of a synthesis route, the exact action steps for conducting the synthesis have to be extracted in a tedious manner, typically manually by humans from patents. This problem was largely solved in a seminal work by *Vaucher* et al. [30] who presented a method to convert unstructured experimental procedures from patents into a stepwise execution protocol of actions needed to conduct the synthesis physically.

To avoid the slow and costly manual synthesis, robotic hardware for automated synthesis is rapidly emerging. *Burger* et al. [389] developed a mobile robot for the search of photocatalysts, and *Coley* et al. [390] reported a preliminary integration of a platform for automated organic synthesis guided by algorithmically predicted synthesis routes. Later, the `IBM RoboRXN` platform integrated a wide range of synthesis planning tools [28, 30, 60, 61, 391] with a programmable robotic hardware that covers a huge organic reaction space and allows to plan and execute synthesis from the browser[1]. The *Matterlab* has developed a rich array of laboratory automation hardware for material design tasks that has

[1] `https://rxn.res.ibm.com`

frequently been demonstrated to accelerate material discovery, e.g., on applications like thin-film materials [15], metal-organic frameworks [16], nanoparticle synthesis [392], process optimization [393] or organic photovoltaics [394].

Outlook

Altogether, within this thesis we hope to have proposed a stack of methods that can flexibly generate molecules for a wide range of complex starting conditions without the need of finetuning.
From the ML viewpoint we believe that our work is in line with the current trends in NLP toward multitask models and prompt design [378, 379, 395]. The next frontier will be to develop foundation models in chemistry that can excel across the full stack of chemoinformatics tasks with little to no necessary adaption. By bridging chemical and natural languages, we envision a model that can be prompted by a chemist with a natural text query describing the design problem and that generates a molecule addressing the design task. Attempts in this direction have been made with the `MolT5` model by *Edwards* et al. [396] on converting natural text describing molecules to SMILES[2]. By flipping the translation task, such a model can be used for tasks like molecule captioning (i.e., natural language descriptions of molecules) which could be particularly interesting to obtain further information about a *de novo* molecule prior to the first ever synthesis (for such molecules this information will likely not be manually retrievable from PubChem). A unified chemical language model called `T5Chem` has been proposed by *Lu* and *Zhang* [335]. `T5Chem` achieves competitive performance on forward reaction prediction, single-step retrosynthesis, reaction classification and reaction yield prediction. However, that model is only a mulitask but not a multidomain model and still relies on task-specific heads. A multidomain model for chemistry-related question-answering, named entity recognition or relation extraction has been explored by *Zeng* et al. [397]. In sum this indicates the beginning of a trend toward multitask/multidomain chemical language models that might culminate in a chemistry foundation model similar to T5 [398], GPT3 [379] or PaLM [399] in NLP.

From a chemist's viewpoint, the next frontier would be the successful deployment of our developed methodology into a fully autonomous DMTA cycle. While we have demonstrated the integration of our algorithms with synthesis planning tools and robotic hardware for autonomous synthesis in Section 4.4, we have not obtained any experimental validation. Ideally, this could be achieved with automatized high-throughput screening like in Arctoris' `Ulysses` platform for kinase inhibitor characterization [400]. If such a platform could be coupled with the models developed in this thesis and the synthesis

[2] Example input: *"The molecule is a sulfonated xanthene dye of absorption wavelength 573 nm and emission wavelength 591 nm. It has a role as a fluorochrome." [396]*

planning and execution tools in `IBM RXN for Chemistry`, one could envision a DMTA cycle where all critical decisions (*what* compound to make next – and *how*?) are being made by machines and all main operations (synthesis and screening assay) are performed without human labor.

Appendix

A1 Introduction

A2 Molecular Property Prediction

Dataset # of tasks	BACE 1	BBBP 1	Tox21 12	Clintox 2	SIDER 27	**Average**
ToxSmi	$0.861_{\pm 0.039}$	$\underline{0.915}_{\pm 0.023}$	$0.795_{\pm 0.050}$	$0.896_{\pm 0.006}$	$0.619_{\pm 0.037}$	$0.817_{\pm 0.031}$
TF_Robust [104]	$0.824_{\pm 0.022}$	$0.860_{\pm 0.087}$	$0.698_{\pm 0.012}$	$0.765_{\pm 0.085}$	$0.607_{\pm 0.033}$	$0.751_{\pm 0.048}$
GraphConv [105]	$0.854_{\pm 0.011}$	$0.877_{\pm 0.036}$	$0.772_{\pm 0.041}$	$0.845_{\pm 0.051}$	$0.593_{\pm 0.035}$	$0.788_{\pm 0.03}$
Weave [106]	$0.791_{\pm 0.008}$	$0.837_{\pm 0.065}$	$0.741_{\pm 0.044}$	$0.823_{\pm 0.023}$	$0.543_{\pm 0.034}$	$0.747_{\pm 0.035}$
SchNet [107]	$0.750_{\pm 0.033}$	$0.847_{\pm 0.024}$	$0.767_{\pm 0.025}$	$0.717_{\pm 0.042}$	$0.545_{\pm 0.038}$	$0.725_{\pm 0.032}$
MPNN [85]	$0.815_{\pm 0.044}$	$0.913_{\pm 0.041}$	$0.808_{\pm 0.024}$	$0.879_{\pm 0.054}$	$0.595_{\pm 0.030}$	$0.802_{\pm 0.04}$
MGCN [109]	$0.734_{\pm 0.030}$	$0.850_{\pm 0.064}$	$0.707_{\pm 0.016}$	$0.634_{\pm 0.042}$	$0.552_{\pm 0.018}$	$0.695_{\pm 0.034}$
AttentiveFP [37]	$0.863_{\pm 0.015}$	$0.908_{\pm 0.050}$	$0.807_{\pm 0.020}$	$\mathbf{0.933}_{\pm 0.020}$	$0.605_{\pm 0.060}$	$\underline{0.823}_{\pm 0.033}$
N-GRAM [110]	$\underline{0.876}_{\pm 0.035}$	$0.912_{\pm 0.013}$	$0.769_{\pm 0.027}$	$0.855_{\pm 0.037}$	$\underline{0.632}_{\pm 0.005}$	$0.808_{\pm 0.023}$
Hu et al. [108]	$0.851_{\pm 0.027}$	$\underline{0.915}_{\pm 0.040}$	$\underline{0.811}_{\pm 0.015}$	$0.762_{\pm 0.058}$	$0.614_{\pm 0.006}$	$0.791_{\pm 0.029}$
GROVER [36]	$\mathbf{0.878}_{\pm 0.016}$	$\mathbf{0.936}_{\pm 0.008}$	$\mathbf{0.819}_{\pm 0.020}$	$\underline{0.925}_{\pm 0.013}$	$\mathbf{0.656}_{\pm 0.006}$	$\mathbf{0.843}_{\pm 0.013}$

Table A2.1: ROC-AUC values for different algorithms evaluated on MoleculeNet datasets split using a *scaffold* splitting strategy. For each dataset the average ROC-AUC across the tasks is reported. Results for ToxSmi were obtained by measuring test performance for 10 repeated scaffold splits. All other numbers are taken from *Rong* et al. [36] who trained all models on 3 repeated scaffold splits.

A3 Proteochemometrics

Data	Config	k-NN	BiMCA	BiMCA (pretrained)
Val.	Full seq.	1.34 ± 0.16	1.38 ± 0.08	1.30 ± 0.13
	Active site	1.32 ± 0.17	$\mathbf{1.28} \pm 0.13$	$\mathbf{1.21} \pm 0.13$
Test	Full seq.	1.56 ± 0.09	1.44 ± 0.04	1.32 ± 0.04
	Active site	$\mathbf{1.52} \pm 0.10$	$\mathbf{1.33} \pm 0.04$	$\mathbf{1.25} \pm 0.05$

Table A3.1: RMSE (on pIC50) on validation and test data (kinase split).

Data	Config	k-NN	BiMCA	BiMCA (pretrained)
Val.	Full seq.	0.41 ± 0.09	0.32 ± 0.05	0.39 ± 0.08
	Active site	$\mathbf{0.42} \pm 0.11$	$\mathbf{0.46} \pm 0.08$	$\mathbf{0.49} \pm 0.07$
Test.	Full seq.	0.23 ± 0.05	0.32 ± 0.03	0.43 ± 0.03
	Active site	$\mathbf{0.28} \pm 0.06$	$\mathbf{0.44} \pm 0.04$	$\mathbf{0.49} \pm 0.05$

Table A3.2: Pearson correlation coefficient on validation and test data (kinase split).

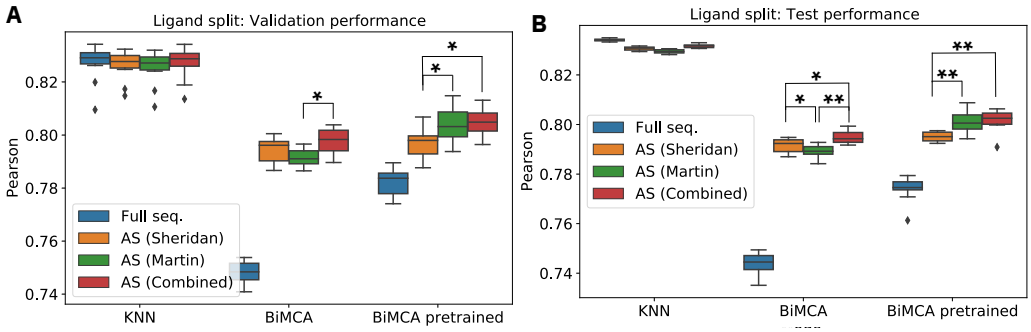


Figure A3.1: Pearson correlation in affinity prediction on the ligand split. Results on validation (**A**) and test data (**B**) are shown. Statistically significant differences between the three different active-site configurations are marked with a star.

Split	Config	Encoding	Validation		Test	
			RMSE	PCC	RMSE	PCC
Kinase	Full seq.	One-hot	$1.41_{\pm 0.08}$	$0.29_{\pm 0.08}$	$\mathbf{1.43}_{\pm 0.05}$	$\mathbf{0.32}_{\pm 0.05}$
Kinase	Full seq.	BLOSUM62	$1.39_{\pm 0.14}$	$\mathbf{0.33}_{\pm 0.06}$	$1.45_{\pm 0.04}$	$0.28_{\pm 0.04}$
Kinase	Full seq.	Learned	$\mathbf{1.38}_{\pm 0.08}$	$0.32_{\pm 0.05}$	$1.44_{\pm 0.04}$	$\mathbf{0.32}_{\pm 0.03}$
Kinase	Active site	One-hot	$1.33_{\pm 0.14}$	$0.44_{\pm 0.08}$	$1.36_{\pm 0.08}$	$\mathbf{0.44}_{\pm 0.05}$
Kinase	Active site	BLOSUM62	$1.31_{\pm 0.13}$	$0.44_{\pm 0.08}$	$1.34_{\pm 0.05}$	$\mathbf{0.44}_{\pm 0.03}$
Kinase	Active site	Learned	$\mathbf{1.28}_{\pm 0.13}$	$\mathbf{0.46}_{\pm 0.08}$	$\mathbf{1.33}_{\pm 0.04}$	$\mathbf{0.44}_{\pm 0.04}$
Ligand	Full seq.	One-hot	$0.91_{\pm 0.01}$	$\mathbf{0.75}_{\pm 0.00}$	$\mathbf{0.91}_{\pm 0.01}$	$\mathbf{0.75}_{\pm 0.00}$
Ligand	Full seq.	BLOSUM62	$0.90_{\pm 0.01}$	$\mathbf{0.75}_{\pm 0.01}$	$\mathbf{0.91}_{\pm 0.01}$	$\mathbf{0.75}_{\pm 0.00}$
Ligand	Full seq.	Learned	$\mathbf{0.91}_{\pm 0.01}$	$\mathbf{0.75}_{\pm 0.00}$	$\mathbf{0.91}_{\pm 0.01}$	$0.74_{\pm 0.00}$
Ligand	Active site	One-hot	$\mathbf{0.83}_{\pm 0.01}$	$\mathbf{0.80}_{\pm 0.00}$	$\mathbf{0.84}_{\pm 0.01}$	$\mathbf{0.79}_{\pm 0.00}$
Ligand	Active site	BLOSUM62	$\mathbf{0.83}_{\pm 0.01}$	$0.79_{\pm 0.01}$	$\mathbf{0.84}_{\pm 0.01}$	$\mathbf{0.79}_{\pm 0.01}$
Ligand	Active site	Learned	$\mathbf{0.83}_{\pm 0.01}$	$0.79_{\pm 0.00}$	$0.83_{\pm 0.01}$	$\mathbf{0.79}_{\pm 0.00}$

Table A3.3: Validation data performance of ablation study on different AA encodings for BiMCA model. Within each split, data partition and metric, the best encoding type is marked in bold.

		RMSE			**Pearson**		
Data	Config	BiMCA	BiMCA-p.	k-NN	BiMCA	BiMCA-pre	k-NN
Val.	Sheridan	$\mathbf{1.28}_{\pm 0.13}$	$1.21_{\pm 0.13}$	$\mathbf{1.32}_{\pm 0.17}$	$0.456_{\pm 0.07}$	$0.487_{\pm 0.07}$	$\mathbf{0.422}_{\pm 0.11}$
	Martin	$1.30_{\pm 0.16}$	$1.22_{\pm 0.13}$	$1.36_{\pm 0.18}$	$\mathbf{0.461}_{\pm 0.07}$	$0.481_{\pm 0.09}$	$0.397_{\pm 0.11}$
	Combined	$1.32_{\pm 0.16}$	$\mathbf{1.20}_{\pm 0.12}$	$1.33_{\pm 0.18}$	$0.438_{\pm 0.08}$	$\mathbf{0.489}_{\pm 0.09}$	$0.419_{\pm 0.11}$
Test	Sheridan	$\mathbf{1.33}_{\pm 0.04}$	$1.25_{\pm 0.05}$	$1.52_{\pm 0.10}$	$0.437_{\pm 0.04}$	$0.488_{\pm 0.05}$	$0.276_{\pm 0.06}$
	Martin	$1.34_{\pm 0.08}$	$1.24_{\pm 0.07}$	$\mathbf{1.48}_{\pm 0.11}$	$\mathbf{0.450}_{\pm 0.05}$	$\mathbf{0.509}_{\pm 0.05}$	$\mathbf{0.296}_{\pm 0.06}$
	Combined	$\mathbf{1.33}_{\pm 0.08}$	$\mathbf{1.23}_{\pm 0.08}$	$1.51_{\pm 0.10}$	$0.431_{\pm 0.06}$	$0.505_{\pm 0.07}$	$0.262_{\pm 0.06}$

Table A3.4: Results on validation and test data (kinase split). For each model and data partition we mark the better representation in bold.

A4 Conditional Molecular Generative Models

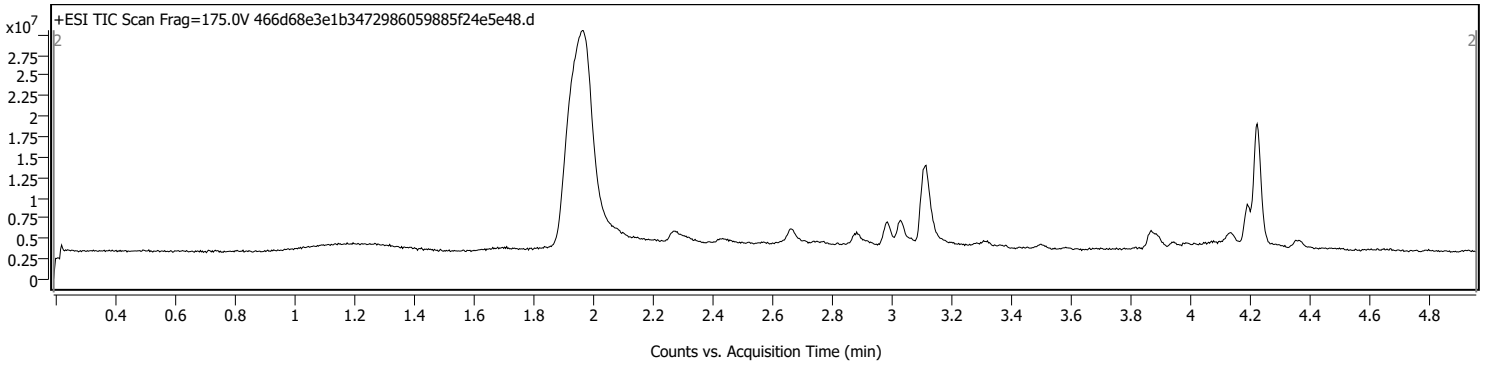
Target Screening Report

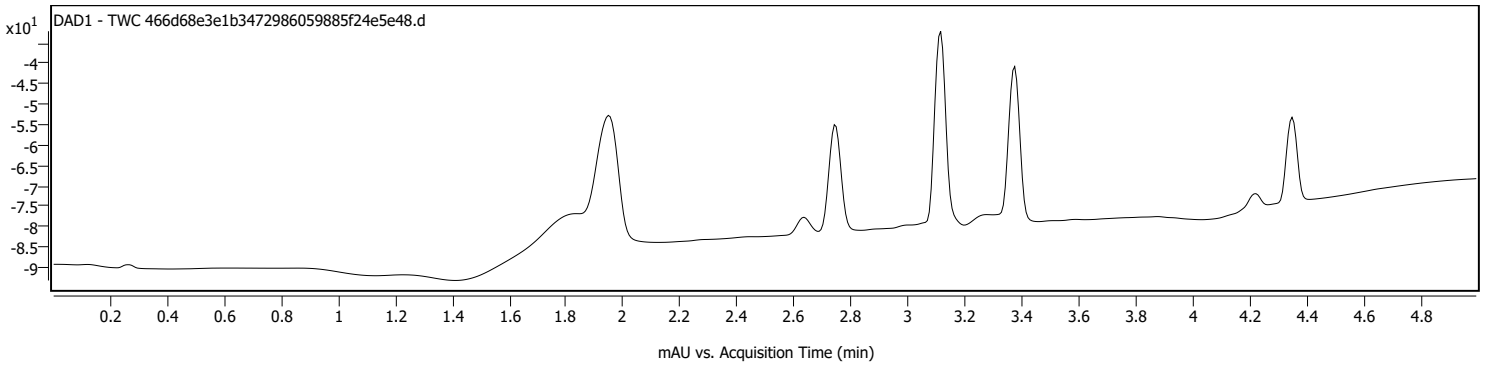
Agilent | Trusted Answers

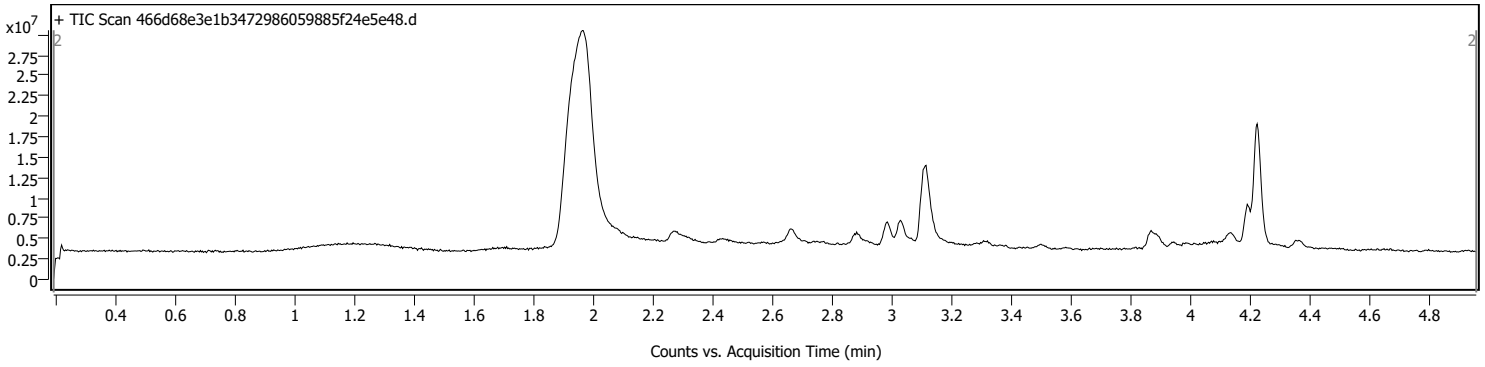
Sample Information

Name	466d68e3e1b3472986059885f24e5e48	**Data File Path**	D:\MassHunter\ChemspeedData\466d68e3e1b3472986059885f24e5e48.d
Sample ID	466d68e3e1b3472986059885f24e5e48	**Acq. Time (Local)**	25/08/2020 22:55:15 (UTC+02:00)
Instrument	RoboRXN	**Method Path (Acq)**	D:\MassHunter\Methods\agilent_ChemspeedValve+QualAnaly.m
MS Type	TOF	**Version (Acq SW)**	6200 series TOF/6500 series Q-TOF 10.1 (48.0)
Inj. Vol. (ul)	0.005	**IRM Status**	Some ions missed
Position	Vial 1	**Method Path (DA)**	D:\MassHunter\ChemspeedData\466d68e3e1b3472986059885f24e5e48.d\AcqData\MethodDA\agilent_Chemspeed_QualAnaly.m
Plate Pos.		**Target Source Path**	X:\analytes.csv
Operator		**Result Summary**	4 qualified (5 targets)

Sample Chromatograms







Compound Summary

Cpd	Name	Formula	CAS	RT	Mass	Mass (Tgt)	Diff (Tgt, ppm)	Score	Algorithm
1		C7 H8 Br N		2.882	184.9858	184.9840	9.55	90.36	FBF
2		C7 H8 Br N		1.970	184.9837	184.9840	-1.64	99.44	FBF
3		C4 H8 O		3.515	72.0580	72.0575	6.74	83.13	FBF
4		C4 H8 O		3.439	72.0571	72.0575	-6.36	80.60	FBF

Compound Details

Cpd. 1: C7 H8 Br N

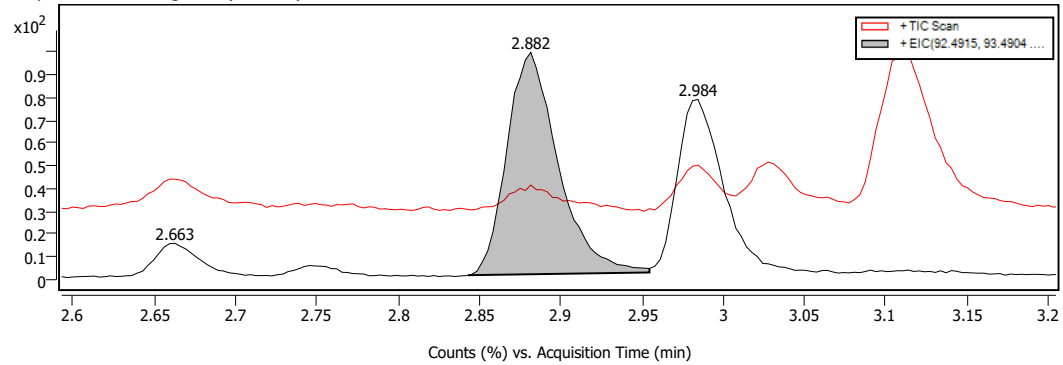
Name	Formula	RT	RI	Mass	Diff (Tgt, ppm)	CAS	ID Source	Score	Algorithm
	C7 H8 Br N	2.882		184.9858	9.55		FBF	90.36	FBF

Species	m/z	Score (Tgt)	Score (Lib)	Score (DB)	Score (MFG)	Score (RT)
(M+H)+ (2M+Na)+	185.9902 392.9582	90.36				

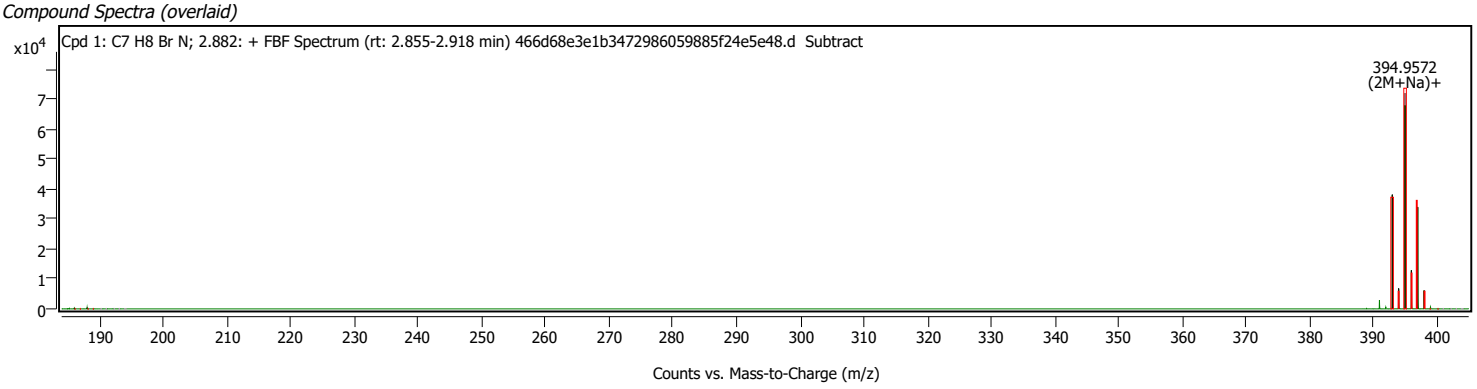
Target Screening Report

Compound Chromatograms (overlaid)

Structure



Compound Spectra (overlaid)



Compound ID Table

Name	Formula	Species	RT	RT Diff	Mass	CAS	ID Source	Score	Score (Lib)	Score (Tgt)
	C7 H8 Br N	(M+H)+ (2M+Na)+	2.882		184.9858		FBF	90.36		90.36

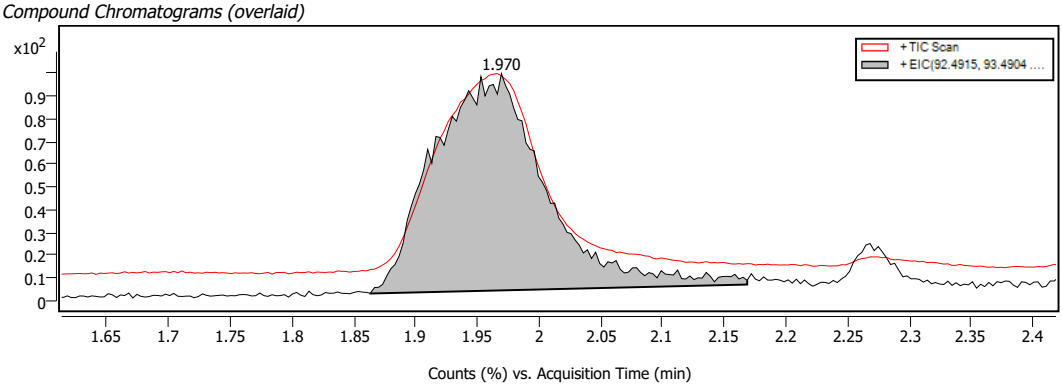
Cpd. 2: C7 H8 Br N

Name	Formula	RT	RI	Mass	Diff (Tgt, ppm)	CAS	ID Source	Score	Algorithm
	C7 H8 Br N	1.970		184.9837	-1.64		FBF	99.44	FBF

Species	m/z	Score (Tgt)	Score (Lib)	Score (DB)	Score (MFG)	Score (RT)
(M+H)+	185.9910	99.44				

Compound Chromatograms (overlaid)

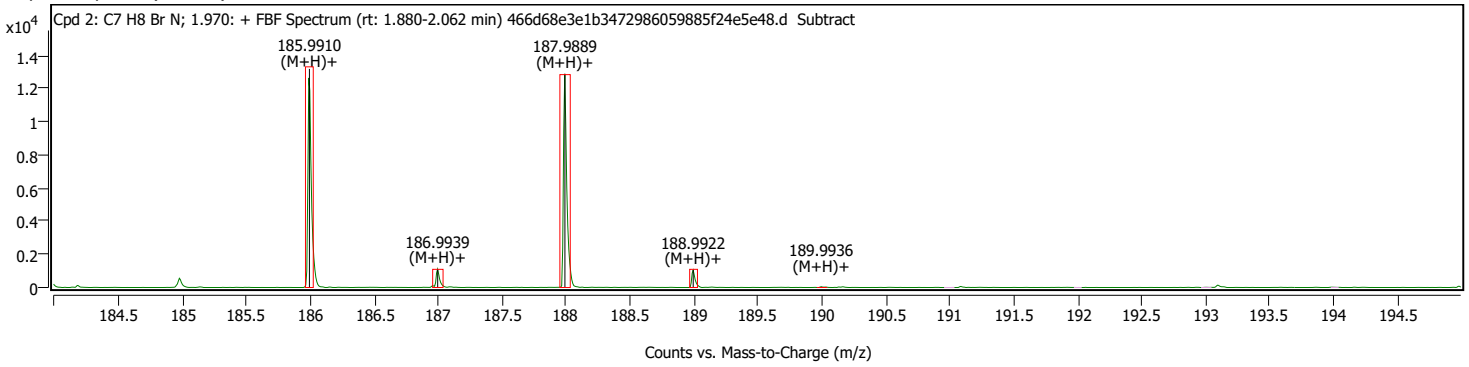
Structure



Target Screening Report

Agilent | Trusted Answers

Compound Spectra (overlaid)



Compound ID Table

Name	Formula	Species	RT	RT Diff	Mass	CAS	ID Source	Score	Score (Lib)	Score (Tgt)
	C7 H8 Br N	(M+H)+	1.970		184.9837		FBF	99.44		99.44

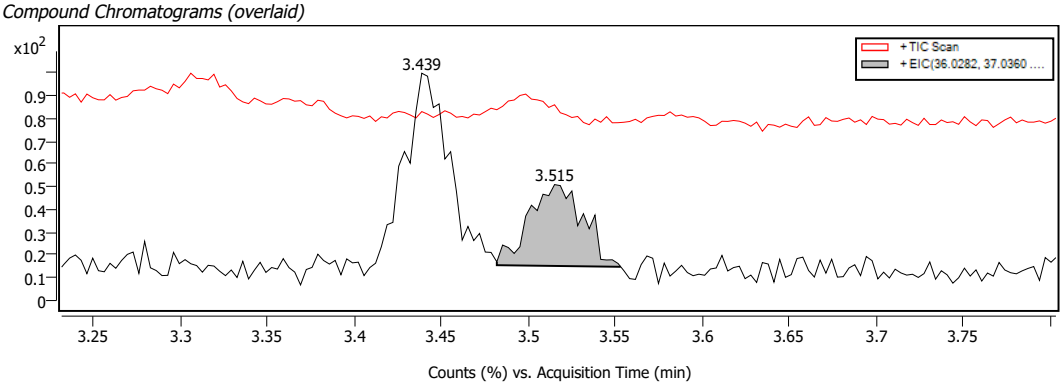
Cpd. 3: C4 H8 O

Name	Formula	RT	RI	Mass	Diff (Tgt, ppm)	CAS	ID Source	Score	Algorithm
	C4 H8 O	3.515		72.0580	6.74		FBF	83.13	FBF

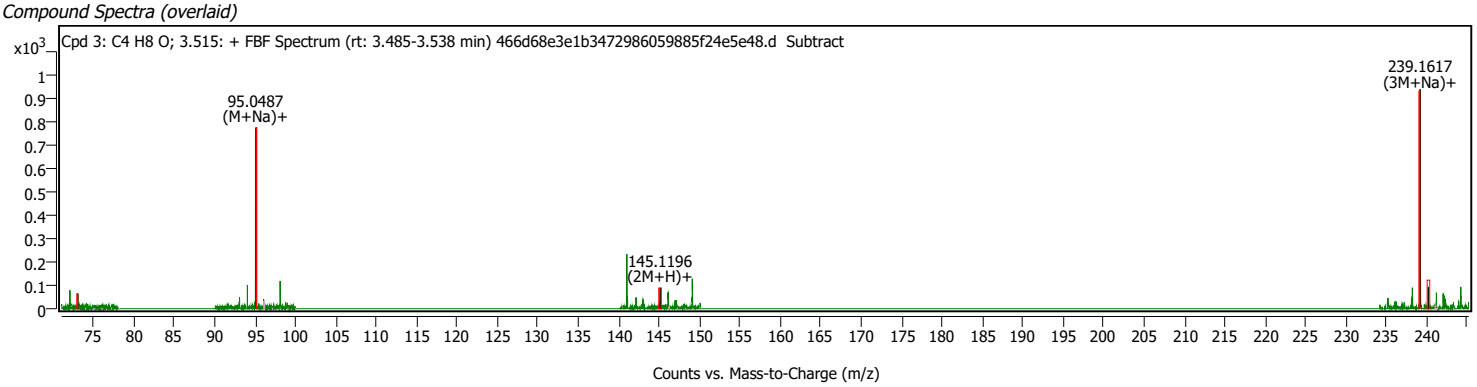
Species	m/z	Score (Tgt)	Score (Lib)	Score (DB)	Score (MFG)	Score (RT)
(M+H)+ (M+Na)+ (2M+H)+ (3M+Na)+	73.0629 95.0487 145.1196 239.1617	83.13				

Compound Chromatograms (overlaid)

Structure



Compound Spectra (overlaid)



Compound ID Table

Name	Formula	Species	RT	RT Diff	Mass	CAS	ID Source	Score	Score (Lib)	Score (Tgt)
	C4 H8 O	(M+H)+ (M+Na)+ (2M+H)+ (3M+Na)+	3.515		72.0580		FBF	83.13		83.13

Cpd. 4: C4 H8 O

Name	Formula	RT	RI	Mass	Diff (Tgt, ppm)	CAS	ID Source	Score	Algorithm
	C4 H8 O	3.439		72.0571	-6.36		FBF	80.60	FBF

Species	m/z	Score (Tgt)	Score (Lib)	Score (DB)	Score (MFG)	Score (RT)
(2M+Na)+ (3M+Na)+	167.1063 239.1611	80.60				

A5 Bridging Property Prediction and Conditional Generation

Metric	$\alpha = 0$, no FE	$\alpha = \mathbf{1}$, no FE	$\alpha = \mathbf{0}$, with FE	$\alpha = \mathbf{1}$, with FE
0-Variance (↓)	$\mathbf{4.4}_{\pm 0.8}$	$5.9_{\pm 1.3}$	$6.1_{\pm 3.7}$	$6.1_{\pm 1.5}$
Spearman ρ (RT)	$0.38_{\pm 0.1}$	$0.38_{\pm 0.0}$	$0.41_{\pm 0.1}$	$\mathbf{0.44}_{\pm 0.0}$
Spearman ρ (Grover)	$0.44_{\pm 0.0}$	$0.46_{\pm 0.0}$	$0.46_{\pm 0.1}$	$\mathbf{0.47}_{\pm 0.0}$

(a) ESOL

Table A5.1: Conditional generation for MoleculeNet datasets. Average performances across all splits for training with alternating objectives are given. "Spearman ρ with RT" refers to the self-evaluation whereas "ρ with Grover" refers to to predictions obtained with the Grover model from *Rong* et al. [36].

Acronyms

ATP	Adenosine triphosphate
AUC	Area under the curve
CASP	Computer-aided synthesis planning
CNN	Convolutional neural network
CV	Cross validation
DL	Deep learning
DMTA	Design-make-test-analyze
DTI	Drug-target interaction
ECFP	Extended-connectivity fingerprint
ELBO	Evidence lower bound
FDA	Food and drug administration
GAN	Generative adversarial network
GCN	Graph convolutional neural network
GEP	Gene expression profile
GNN	Graph neural network
GP	Gaussian process
GRU	Gated Recurrent Unit
HTS	High-throughput screening
LOOCV	Leave-one-out cross-validation
MLM	Masked language modeling
MOA	Mechanism of action
MSA	Multiple sequence alignment
MSE	Mean-squared error
NLP	Natural language processing
PCC	Pearson correlation coefficient
PKA	Protein kinase A
PLM	Permutation language modeling
QED	Quantitative estimate of drug-likeness
QSAR	Quantitative structure-activity relationship
RL	Reinforcement learning
RMSE	Root-mean-square error
RNN	Recurrent neural network
ROC	Receiver operating characteristic

ROC-AUC	Area under the ROC curve
SAS	Synthetic accessibility score
SCScore	Synthetic complexity score
SELFIES	Self-referencing embedded strings
SEM	Standard error of the mean
SMARTS	SMILES arbitrary target specification
SMILES	Simplified molecular-input line-entry system
SNR	Signal-to-noise ratio
SOTA	State-of-the-art
TAPE	Tasks assessing protein embeddings
U	Mann-Whitney U test
VAE	Variational autoencoder
W+	Wilcoxon signed-rank test

Bibliography

1. J. W. *Scannell*, A. *Blanckley*, H. *Boldon*, and B. *Warrington*. "Diagnosing the decline in pharmaceutical R&D efficiency". *Nature reviews Drug discovery* 11:3, 2012, p. 191.
2. J. A. *DiMasi*, H. G. *Grabowski*, and R. W. *Hansen*. "Innovation in the pharmaceutical industry: new estimates of R&D costs". *Journal of health economics* 47, 2016, pp. 20–33.
3. M. S. *Kinch*, A. *Haynesworth*, S. L. *Kinch*, and D. *Hoyer*. "An overview of FDA-approved new molecular entities: 1827–2013". *Drug discovery today* 19:8, 2014, pp. 1033–1039.
4. S. *Kim*, P. A. *Thiessen*, E. E. *Bolton*, J. *Chen*, G. *Fu*, A. *Gindulyte*, L. *Han*, J. *He*, S. *He*, B. A. *Shoemaker*, et al. "PubChem substance and compound databases". *Nucleic acids research* 44:D1, 2016, pp. D1202–D1213.
5. E. *Hargrave-Thomas*, B. *Yu*, and J. *Reynisson*. "Serendipity in anticancer drug discovery". *World journal of clinical oncology* 3:1, 2012, p. 1.
6. D. G. *Brown*, H. J. *Wobst*, A. *Kapoor*, L. A. *Kenna*, and N. *Southall*. "Clinical development times for innovative drugs." *Nature reviews. Drug discovery*, 2021.
7. P. G. *Polishchuk*, T. I. *Madzhidov*, and A. *Varnek*. "Estimation of the size of drug-like chemical space based on GDB-17 data". *Journal of computer-aided molecular design* 27:8, 2013, pp. 675–679.
8. D. P. *Tabor*, L. M. *Roch*, S. K. *Saikin*, C. *Kreisbeck*, D. *Sheberla*, J. H. *Montoya*, S. *Dwaraknath*, M. *Aykol*, C. *Ortiz*, H. *Tribukait*, et al. "Accelerating the discovery of materials for clean energy in the era of smart automation". *Nature Reviews Materials* 3:5, 2018, pp. 5–20.
9. P. *Schneider*, W. P. *Walters*, A. T. *Plowright*, N. *Sieroka*, J. *Listgarten*, R. A. *Goodnow*, J. *Fisher*, J. M. *Jansen*, J. S. *Duca*, T. S. *Rush*, et al. "Rethinking drug design in the artificial intelligence era". *Nature Reviews Drug Discovery* 19:5, 2020, pp. 353–364.
10. H. *Chen*, O. *Engkvist*, Y. *Wang*, M. *Olivecrona*, and T. *Blaschke*. "The rise of deep learning in drug discovery". *Drug discovery today*, 2018.

11. R. *Gomez-Bombarelli*, J. N. *Wei*, D. *Duvenaud*, J. M. *Hernandez-Lobato*, et al. "Automatic chemical design using a data-driven continuous representation of molecules". *ACS central science* 4:2, 2018, pp. 268–276.

12. A. *Zhavoronkov*, Y. A. *Ivanenkov*, A. *Aliper*, M. S. *Veselov*, V. A. *Aladinskiy*, A. V. *Aladinskaya*, V. A. *Terentiev*, D. A. *Polykovskiy*, M. D. *Kuznetsov*, A. *Asadulaev*, et al. "Deep learning enables rapid identification of potent DDR1 kinase inhibitors". *Nature biotechnology* 37:9, 2019, pp. 1038–1040.

13. P. *Das*, T. *Sercu*, K. *Wadhawan*, I. *Padhi*, S. *Gehrmann*, F. *Cipcigan*, V. *Chenthamarakshan*, H. *Strobelt*, C. *Dos Santos*, P.-Y. *Chen*, et al. "Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations". *Nature Biomedical Engineering* 5:6, 2021, pp. 613–623.

14. F. *Grisoni*, B. J. *Huisman*, A. L. *Button*, M. *Moret*, K. *Atz*, D. *Merk*, and G. *Schneider*. "Combining generative artificial intelligence and on-chip synthesis for de novo drug design". *Science Advances* 7:24, 2021, eabg3338.

15. B. P. *MacLeod*, F. G. *Parlane*, T. D. *Morrissey*, F. *Häse*, L. M. *Roch*, K. E. *Dettelbach*, R. *Moreira*, L. P. *Yunker*, M. B. *Rooney*, J. R. *Deeth*, et al. "Self-driving laboratory for accelerated discovery of thin-film materials". *Science Advances* 6:20, 2020, eaaz8867.

16. Z. *Yao*, B. *Sánchez-Lengeling*, N. S. *Bobbitt*, B. J. *Bucior*, S. G. H. *Kumar*, S. P. *Collins*, T. *Burns*, T. K. *Woo*, O. K. *Farha*, R. Q. *Snurr*, et al. "Inverse design of nanoporous crystalline reticular materials with deep generative models". *Nature Machine Intelligence* 3:1, 2021, pp. 76–86.

17. D. *Flam-Shepherd*, K. *Zhu*, and A. *Aspuru-Guzik*. "Language models can learn complex molecular distributions". *Nature Communications* 13:1, 2022, pp. 1–10.

18. M. *Popova*, O. *Isayev*, and A. *Tropsha*. "Deep reinforcement learning for de novo drug design". *Science advances* 4:7, 2018, eaap7885.

19. W. *Jin*, R. *Barzilay*, and T. *Jaakkola*. "Junction tree variational autoencoder for molecular graph generation". In: *International conference on machine learning*. PMLR. 2018, pp. 2323–2332.

20. Y. *Kwon*, J. *Yoo*, Y.-S. *Choi*, W.-J. *Son*, D. *Lee*, and S. *Kang*. "Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation". *Journal of Cheminformatics* 11:1, 2019, pp. 1–10.

21. A. *Domenico*, G. *Nicola*, T. *Daniela*, C. *Fulvio*, A. *Nicola*, and N. *Orazio*. "De novo drug design of targeted chemical libraries based on artificial intelligence and pair-based multiobjective optimization". *Journal of Chemical Information and Modeling* 60:10, 2020, pp. 4582–4593.

22. D. *Polykovskiy*, A. *Zhebrak*, D. *Vetrov*, Y. *Ivanenkov*, V. *Aladinskiy*, P. *Mamoshina*, M. *Bozdaganyan*, A. *Aliper*, A. *Zhavoronkov*, and A. *Kadurin*. "Entangled conditional adversarial autoencoder for de novo drug discovery". *Molecular pharmaceutics* 15:10, 2018, pp. 4398–4405.

23. E. *Putin*, A. *Asadulaev*, Q. *Vanhaelen*, Y. *Ivanenkov*, A. V. *Aladinskaya*, A. *Aliper*, and A. *Zhavoronkov*. "Adversarial threshold neural computer for molecular de novo design". *Molecular pharmaceutics* 15:10, 2018, pp. 4386–4397.

24. C. *Shi*, M. *Xu*, Z. *Zhu*, W. *Zhang*, M. *Zhang*, and J. *Tang*. "GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation". In: *8th International Conference on Learning Representations, ICLR*. 2020.

25. J. *You*, B. *Liu*, Z. *Ying*, V. *Pande*, and J. *Leskovec*. "Graph convolutional policy network for goal-directed molecular graph generation". In: *Advances in neural information processing systems*. 2018, pp. 6410–6421.

26. M. *Manica*, J. *Cadow*, D. *Christofidellis*, A. *Dave*, J. *Born*, D. *Clarke*, Y. G. N. *Teukam*, S. C. *Hoffman*, M. *Buchan*, V. *Chenthamarakshan*, et al. "GT4SD: Generative Toolkit for Scientific Discovery". *arXiv preprint arXiv:2207.03928*, 2022. URL: https://github.com/GT4SD/gt4sd-core.

27. M. *Manica*, A. *Oskooei*, J. *Born*, V. *Subramanian*, J. *Saez-Rodriguez*, and M. *Rodriguez Martinez*. "Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders". *Molecular pharmaceutics* 16:12, 2019, pp. 4797–4806.

28. P. *Schwaller*, T. *Laino*, T. *Gaudin*, P. *Bolgar*, C. A. *Hunter*, C. *Bekas*, and A. A. *Lee*. "Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction". *ACS central science* 5:9, 2019, pp. 1572–1583.

29. P. *Schwaller*, B. *Hoover*, J.-L. *Reymond*, H. *Strobelt*, and T. *Laino*. "Extraction of organic chemistry grammar from unsupervised learning of chemical reactions". *Science Advances* 7:15, 2021, eabe4166.

30. A. C. *Vaucher*, F. *Zipoli*, J. *Geluykens*, V. H. *Nair*, P. *Schwaller*, and T. *Laino*. "Automated extraction of chemical synthesis actions from experimental procedures". *Nature Communications*, 2020.

31. I. *Kola* and J. *Landis*. "Can the pharmaceutical industry reduce attrition rates?" *Nature reviews Drug discovery* 3:8, 2004, pp. 711–716.

32. P. K. *Singh*, A. *Negi*, P. K. *Gupta*, M. *Chauhan*, and R. *Kumar*. "Toxicophore exploration as a screening technology for drug design and discovery: techniques, scope and limitations". *Archives of toxicology* 90:8, 2016, pp. 1785–1802.

33. C. H. *Wong*, K. W. *Siah*, and A. W. *Lo*. "Estimation of clinical trial success rates and related parameters". *Biostatistics* 20:2, 2019, pp. 273–286.

34. Z. *Wu*, B. *Ramsundar*, E. N. *Feinberg*, J. *Gomes*, C. *Geniesse*, A. S. *Pappu*, K. *Leswing*, and V. *Pande*. "MoleculeNet: a benchmark for molecular machine learning". *Chemical science* 9:2, 2018, pp. 513–530.

35. K. *Yang*, K. *Swanson*, W. *Jin*, C. *Coley*, P. *Eiden*, H. *Gao*, A. *Guzman-Perez*, T. *Hopper*, B. *Kelley*, M. *Mathea*, et al. "Analyzing learned molecular representations for property prediction". *Journal of chemical information and modeling* 59:8, 2019, pp. 3370–3388.

36. Y. *Rong*, Y. *Bian*, T. *Xu*, W. *Xie*, Y. *Wei*, W. *Huang*, and J. *Huang*. "Self-supervised graph transformer on large-scale molecular data". *Advances in Neural Information Processing Systems* 33, 2020, pp. 12559–12571.

37. Z. *Xiong*, D. *Wang*, X. *Liu*, F. *Zhong*, X. *Wan*, X. *Li*, Z. *Li*, X. *Luo*, K. *Chen*, H. *Jiang*, et al. "Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism". *Journal of medicinal chemistry* 63:16, 2019, pp. 8749–8760.

38. T. B. *Kimber*, M. *Gagnebin*, and A. *Volkamer*. "Maxsmi: maximizing molecular property prediction performance with confidence estimation using SMILES augmentation and deep learning". *Artificial Intelligence in the Life Sciences* 1, 2021, p. 100014.

39. A. *Mayr*, G. *Klambauer*, T. *Unterthiner*, and S. *Hochreiter*. "DeepTox: toxicity prediction using deep learning". *Frontiers in Environmental Science* 3, 2016, p. 80.

40. C. *Ji*, F. *Svensson*, A. *Zoufir*, and A. *Bender*. "EMolTox: prediction of molecular toxicity with confidence". *Bioinformatics* 34:14, 2018, pp. 2508–2509.

41. H. *Yang*, L. *Sun*, W. *Li*, G. *Liu*, and Y. *Tang*. "In silico prediction of chemical toxicity for drug design using machine learning methods and structural alerts". *Frontiers in chemistry* 6, 2018, p. 30.

42. Y. *Peng*, Z. *Zhang*, Q. *Jiang*, J. *Guan*, and S. *Zhou*. "TOP: Towards better toxicity prediction by deep molecular representation learning". In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2019, pp. 318–325.

43. M. *Zaslavskiy*, S. *Jégou*, E. W. *Tramel*, and G. *Wainrib*. "ToxicBlend: Virtual screening of toxic compounds with ensemble predictors". *Computational Toxicology* 10, 2019, pp. 81–88.

44. A. *Karim*, A. *Mishra*, M. H. *Newton*, and A. *Sattar*. "Efficient toxicity prediction via simple features using shallow neural networks and decision trees". *Acs Omega* 4:1, 2019, pp. 1874–1888.

45. K. V. *Chuang*, L. *Gunsalus*, and M. J. *Keiser*. "Learning Molecular Representations for Medicinal Chemistry". *Journal of Medicinal Chemistry*, 2020.

46. D. *Rogers* and M. *Hahn*. "Extended-connectivity fingerprints". *Journal of chemical information and modeling* 50:5, 2010, pp. 742–754.

47. J. *Born* and M. *Manica*. "Trends in Deep Learning for Property-driven Drug Design". *Current Medicinal Chemistry* 28:38, 2021, pp. 7862–7886.

48. D. *Weininger*. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". *Journal of chemical information and computer sciences* 28:1, 1988, pp. 31–36.

49. E. N. *Muratov*, J. *Bajorath*, R. P. *Sheridan*, I. V. *Tetko*, D. *Filimonov*, V. *Poroikov*, T. I. *Oprea*, I. I. *Baskin*, A. *Varnek*, A. *Roitberg*, et al. "QSAR without borders". *Chemical Society Reviews*, 2020.

50. A. *Vaswani*, N. *Shazeer*, N. *Parmar*, J. *Uszkoreit*, et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

51. S. *Jastrzebski*, D. *Leśniak*, and W. M. *Czarnecki*. "Learning to SMILE (S)". *arXiv preprint arXiv:1602.06289*, 2016. ICLR 2016 Workshop.

52. H. *Ozturk*, A. *Ozgur*, and E. *Ozkirimli*. "DeepDTA: deep drug–target binding affinity prediction". *Bioinformatics* 34:17, 2018, pp. i821–i829.

53. P. *Schwaller*, T. *Gaudin*, D. *Lanyi*, C. *Bekas*, and T. *Laino*. ""Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models". *Chemical science* 9:28, 2018, pp. 6091–6098.

54. M. *Hirohara*, Y. *Saito*, Y. *Koda*, K. *Sato*, and Y. *Sakakibara*. "Convolutional neural network based on SMILES representation of compounds for detecting chemical motif". *BMC bioinformatics* 19:19, 2018, pp. 83–94.

55. M. H. *Segler*, T. *Kogej*, C. *Tyrchan*, and M. P. *Waller*. "Generating focused molecule libraries for drug discovery with recurrent neural networks". *ACS central science* 4:1, 2018, pp. 120–131.

56. E. J. *Bjerrum*. "SMILES enumeration as data augmentation for neural network modeling of molecules". *arXiv preprint arXiv:1703.07076*, 2017.

57. S. *Kim*, J. *Chen*, T. *Cheng*, A. *Gindulyte*, J. *He*, S. *He*, Q. *Li*, B. A. *Shoemaker*, P. A. *Thiessen*, B. *Yu*, et al. "PubChem 2019 update: improved access to chemical data". *Nucleic acids research*, 2018.

58. G. *Landrum*. "RDKit: Open-source cheminformatics, v. 2019". *GitHub (https://github. com/rdkit/rdkit)*, 2019.

59. N. M. *O'Boyle*. "Towards a Universal SMILES representation-A standard method to generate canonical SMILES based on the InChI". *Journal of cheminformatics* 4:1, 2012, p. 22.

60. P. *Schwaller*, R. *Petraglia*, V. *Zullo*, V. H. *Nair*, R. A. *Haeuselmann*, R. *Pisoni*, C. *Bekas*, A. *Iuliano*, and T. *Laino*. "Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy". *Chemical Science* 11:12, 2020, pp. 3316–3325.

61. A. C. *Vaucher*, P. *Schwaller*, J. *Geluykens*, V. H. *Nair*, A. *Iuliano*, and T. *Laino*. "Inferring experimental procedures from text-based representations of chemical reactions". *Nature communications* 12:1, 2021, pp. 1–11.

62. P. *Schwaller*, A. C. *Vaucher*, T. *Laino*, and J.-L. *Reymond*. "Prediction of chemical reaction yields using deep learning". *Machine learning: science and technology* 2:1, 2021, p. 015016.

63. Y. *Gal* and Z. *Ghahramani*. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *International conference on machine learning (ICML)*. PMLR. 2016, pp. 1050–1059.

64. M. S. *Ayhan* and P. *Berens*. "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks". In: *Proceedings of the 1st Conference on Medical Imaging with Deep Learning (MIDL)*. 2018.

65. M. *Krenn*, F. *Häse*, A. *Nigam*, P. *Friederich*, and A. *Aspuru-Guzik*. "Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation". *Machine Learning: Science and Technology* 1:4, 2020, p. 045024.

66. X. *Li* and D. *Fourches*. "SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning". *Journal of Chemical Information and Modeling* 61:4, 2021, pp. 1560–1569.

67. J. *Arús-Pous*, S. V. *Johansson*, O. *Prykhodko*, E. J. *Bjerrum*, C. *Tyrchan*, J.-L. *Reymond*, H. *Chen*, and O. *Engkvist*. "Randomized SMILES strings improve the quality of molecular generative models". *Journal of cheminformatics* 11:1, 2019, pp. 1–13.

68. P. *Liu*, H. *Li*, S. *Li*, and K.-S. *Leung*. "Improving prediction of phenotypic drug response on cancer cell lines using deep convolutional network". *BMC bioinformatics* 20:1, 2019, p. 408.

69. N. *O'Boyle* and A. *Dalke*. "DeepSMILES: an adaptation of SMILES for use in machine-learning of chemical structures", 2018.

70. T.-S. *Lin*, C. W. *Coley*, H. *Mochigase*, H. K. *Beech*, W. *Wang*, Z. *Wang*, E. *Woods*, S. L. *Craig*, J. A. *Johnson*, J. A. *Kalow*, et al. "BigSMILES: a structurally-based line notation for describing macromolecules". *ACS central science* 5:9, 2019, pp. 1523–1531.

71. P. *Gage*. "A new algorithm for data compression". *C Users Journal* 12:2, 1994, pp. 23–38.

72. J. *Born*, M. *Manica*, A. *Oskooei*, J. *Cadow*, G. *Markert*, and M. R. *Martınez*. "PaccMann^{RL}: De novo generation of hit-like anticancer molecules from transcriptomic data via reinforcement learning". *Iscience* 24:4, 2021, p. 102269.

73. J. *Cadow*, J. *Born*, M. *Manica*, A. *Oskooei*, and M. *Rodriguez Martinez*. "PaccMann: a web service for interpretable anticancer compound sensitivity prediction". *Nucleic acids research* 48:W1, 2020, W502–W508.

74. D. *Bahdanau*, K. *Cho*, and Y. *Bengio*. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. *Bengio* and Y. *LeCun*. 2015.

75. J. *Chung*, C. *Gulcehre*, K. *Cho*, and Y. *Bengio*. "Empirical evaluation of gated recurrent neural networks on sequence modeling". *arXiv preprint arXiv:1412.3555*, 2014.

76. N. *Vecoven*, D. *Ernst*, and G. *Drion*. "A bio-inspired bistable recurrent cell allows for long-lasting memory". *Plos one* 16:6, 2021, e0252676.

77. D. P. *Kingma* and J. *Ba*. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR*. 2015.

78. T. *Tanimoto*. *An elementary mathematical theory of classification and prediction, IBM Report (November, 1958), cited in: G. Salton, Automatic Information Organization and Retrieval*. 1968.

79. D. K. *Duvenaud*, D. *Maclaurin*, J. *Iparraguirre*, R. *Bombarell*, T. *Hirzel*, A. *Aspuru-Guzik*, and R. P. *Adams*. "Convolutional networks on graphs for learning molecular fingerprints". In: *Advances in neural information processing systems*. 2015, pp. 2224–2232.

80. S. V. N. *Vishwanathan*, N. N. *Schraudolph*, R. *Kondor*, and K. M. *Borgwardt*. "Graph kernels". *Journal of Machine Learning Research* 11, 2010, pp. 1201–1242.

81. K. M. *Borgwardt* and H.-P. *Kriegel*. "Shortest-path kernels on graphs". In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE. 2005, 8–pp.

82. R. W. *Floyd*. "Algorithm 97: shortest path". *Communications of the ACM* 5:6, 1962, p. 345.

83. N. *Shervashidze*, P. *Schweitzer*, E. J. *Van Leeuwen*, K. *Mehlhorn*, and K. M. *Borgwardt*. "Weisfeiler-Lehman graph kernels". *Journal of Machine Learning Research* 12, 2011, pp. 2539–2561.

84. G. *Nikolentzos* and M. *Vazirgiannis*. "Message passing graph kernels". *arXiv preprint arXiv:1808.02510*, 2018.

85. J. *Gilmer*, S. S. *Schoenholz*, P. F. *Riley*, O. *Vinyals*, and G. E. *Dahl*. "Neural message passing for quantum chemistry". In: *International conference on machine learning*. PMLR. 2017, pp. 1263–1272.

86. M. *Togninalli*, E. *Ghisu*, F. *Llinares-López*, B. *Rieck*, and K. *Borgwardt*. "Wasserstein weisfeiler-lehman graph kernels". *Advances in Neural Information Processing Systems* 32, 2019.

87. M. A. *Hearst*, S. T. *Dumais*, E. *Osuna*, J. *Platt*, and B. *Scholkopf*. "Support vector machines". *IEEE Intelligent Systems and their applications* 13:4, 1998, pp. 18–28.

88. R. *Huang*, M. *Xia*, D.-T. *Nguyen*, T. *Zhao*, S. *Sakamuru*, J. *Zhao*, S. A. *Shahane*, A. *Rossoshek*, and A. *Simeonov*. "Tox21Challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs". *Frontiers in Environmental Science* 3, 2016, p. 85.

89. R. R. *Tice*, C. P. *Austin*, R. J. *Kavlock*, and J. R. *Bucher*. "Improving the human hazard characterization of chemicals: a Tox21 update". *Environmental health perspectives* 121:7, 2013, pp. 756–765.

90. G. *Subramanian*, B. *Ramsundar*, V. *Pande*, and R. A. *Denny*. "Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches". *Journal of chemical information and modeling* 56:10, 2016, pp. 1936–1949.

91. M. *Kuhn*, I. *Letunic*, L. J. *Jensen*, and P. *Bork*. "The SIDER database of drugs and side effects". *Nucleic acids research* 44:D1, 2016, pp. D1075–D1079.

92. H. *Altae-Tran*, B. *Ramsundar*, A. S. *Pappu*, and V. *Pande*. "Low data drug discovery with one-shot learning". *ACS central science* 3:4, 2017, pp. 283–293.

93. K. M. *Gayvert*, N. S. *Madhukar*, and O. *Elemento*. "A data-driven approach to predicting successes and failures of clinical trials". *Cell chemical biology* 23:10, 2016, pp. 1294–1301.

94. A. V. *Artemov*, E. *Putin*, Q. *Vanhaelen*, A. *Aliper*, I. V. *Ozerov*, and A. *Zhavoronkov*. "Integrated deep learned transcriptomic and structure-based predictor of clinical trials outcomes". *BioRxiv*, 2016, p. 095653.

95. I. F. *Martins*, A. L. *Teixeira*, L. *Pinheiro*, and A. O. *Falcao*. "A Bayesian approach to in silico blood-brain barrier penetration modeling". *Journal of chemical information and modeling* 52:6, 2012, pp. 1686–1697.

96. *AIDS Antiviral Screen Data*. `http://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data`. 2017.

97. M. *Lisurek*, B. *Rupp*, J. *Wichard*, M. *Neuenschwander*, J. P. von *Kries*, R. *Frank*, J. *Rademann*, and R. *Kühne*. "Design of chemical libraries with potentially bioactive molecules applying a maximum common substructure concept". *Molecular diversity* 14:2, 2010, pp. 401–408.

98. H. E. *Webel*, T. B. *Kimber*, S. *Radetzki*, M. *Neuenschwander*, M. *Nazaré*, and A. *Volkamer*. "Revealing cytotoxic substructures in molecules using deep learning". *Journal of computer-aided molecular design* 34:7, 2020, pp. 731–746.

99. L. *Bennett*, B. *Melchers*, and B. *Proppe*. *Curta: A General-purpose High-Performance Computer at ZEDAT, Freie Universität Berlin*. http://dx.doi.org/10.17169/refubium-26754. 2020.

100. N. M. *Kriege*, F. D. *Johansson*, and C. *Morris*. "A survey on graph kernels". *Applied Network Science* 5:1, 2020, pp. 1–42.

101. M. *Withnall*, E. *Lindelöf*, O. *Engkvist*, and H. *Chen*. "Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction". *Journal of cheminformatics* 12:1, 2020, pp. 1–18.

102. W. X. *Shen*, X. *Zeng*, F. *Zhu*, C. *Qin*, Y. *Tan*, Y. Y. *Jiang*, Y. Z. *Chen*, et al. "Out-of-the-box deep learning prediction of pharmaceutical properties by broadly learned knowledge-based molecular representations". *Nature Machine Intelligence* 3:4, 2021, pp. 334–343.

103. S. *Schroedl*. "Current methods and challenges for deep learning in drug discovery". *Drug Discovery Today: Technologies* 32, 2019, pp. 9–17.

104. B. *Ramsundar*, P. *Eastman*, P. *Walters*, and V. *Pande*. *Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more*. O'Reilly Media, 2019.

105. T. N. *Kipf* and M. *Welling*. "Semi-supervised classification with graph convolutional networks". *arXiv preprint arXiv:1609.02907*, 2016.

106. S. *Kearnes*, K. *McCloskey*, M. *Berndl*, V. *Pande*, and P. *Riley*. "Molecular graph convolutions: moving beyond fingerprints". *Journal of computer-aided molecular design* 30:8, 2016, pp. 595–608.

107. K. *Schütt*, P.-J. *Kindermans*, H. E. *Sauceda Felix*, S. *Chmiela*, A. *Tkatchenko*, and K.-R. *Müller*. "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions". *Advances in neural information processing systems* 30, 2017.

108. W. *Hu*, B. *Liu*, J. *Gomes*, M. *Zitnik*, P. *Liang*, V. S. *Pande*, and J. *Leskovec*. "Strategies for Pre-training Graph Neural Networks". In: *8th International Conference on Learning Representations, ICLR 2020*. 2020.

109. C. *Lu*, Q. *Liu*, C. *Wang*, Z. *Huang*, P. *Lin*, and L. *He*. "Molecular property prediction: A multilevel quantum interactions modeling perspective". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 1052–1060.

110. S. *Liu*, M. F. *Demirel*, and Y. *Liang*. "N-gram graph: Simple unsupervised representation for graphs, with applications to molecules". *Advances in neural information processing systems* 32, 2019.

111. J. *Jiménez-Luna*, F. *Grisoni*, and G. *Schneider*. "Drug discovery with explainable artificial intelligence". *Nature Machine Intelligence* 2:10, 2020, pp. 573–584.

112. Q. *Ding*, S. *Hou*, S. *Zu*, Y. *Zhang*, and S. *Li*. "VISAR: an interactive tool for dissecting chemical features learned by deep neural network QSAR models". *Bioinformatics* 36:11, 2020, pp. 3610–3612.

113. J. *Jiménez-Luna*, M. *Skalic*, N. *Weskamp*, and G. *Schneider*. "Coloring molecules with explainable artificial intelligence for preclinical relevance assessment". *Journal of Chemical Information and Modeling* 61:3, 2021, pp. 1083–1094.

114. J. *Kazius*, R. *McGuire*, and R. *Bursi*. "Derivation and validation of toxicophores for mutagenicity prediction". *Journal of medicinal chemistry* 48:1, 2005, pp. 312–320.

115. H. J. *Verhaar*, C. J. van *Leeuwen*, and J. L. *Hermens*. "Classifying environmental pollutants". *Chemosphere* 25:4, 1992, pp. 471–491. ISSN: 00456535. DOI: 10.1016/0045-6535(92)90280-5.

116. J. L. *Hermens*. "Electrophiles and acute toxicity to fish". In: *Environmental Health Perspectives*. Vol. 87. 1990, pp. 219–225. DOI: 10.1289/ehp.9087219.

117. M. *Nendza*, A. *Wenzel*, M. *Müller*, G. *Lewin*, N. *Simetska*, F. *Stock*, and J. *Arning*. "Screening for potential endocrine disruptors in fish: evidence from structural alerts and in vitro and in vivo toxicological assays". *Environmental Sciences Europe* 28:1, 2016, pp. 1–19.

118. R. *Sayle*. "1st-class SMARTS patterns". In: *EuroMUG 97*. 1997.

119. M. *Karimi*, D. *Wu*, Z. *Wang*, and Y. *Shen*. "DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks". *Bioinformatics* 35:18, 2019, pp. 3329–3338.

120. M. *Tsubaki*, K. *Tomii*, and J. *Sese*. "Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences". *Bioinformatics* 35:2, 2019, pp. 309–318.

121. S. *Li*, F. *Wan*, H. *Shu*, T. *Jiang*, D. *Zhao*, and J. *Zeng*. "MONN: a multi-objective neural network for predicting compound-protein interactions and affinities". *Cell Systems* 10:4, 2020, pp. 308–322.

122. L. *Hirschfeld*, K. *Swanson*, K. *Yang*, R. *Barzilay*, and C. W. *Coley*. "Uncertainty quantification using neural networks for molecular property prediction". *Journal of Chemical Information and Modeling* 60:8, 2020, pp. 3770–3780.

123. I. *Cortés-Ciriano* and A. *Bender*. "Deep confidence: a computationally efficient framework for calculating reliable prediction errors for deep neural networks". *Journal of chemical information and modeling* 59:3, 2018, pp. 1269–1281.

124. I. *Cortes-Ciriano* and A. *Bender*. "Reliable prediction errors for deep neural networks using test-time dropout". *Journal of chemical information and modeling* 59:7, 2019, pp. 3330–3339.

125. A. *Kendall* and Y. *Gal*. "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems* 30, 2017.

126. D. *Baumann* and K. *Baumann*. "Reliable estimation of prediction errors for QSAR models under model uncertainty using double cross-validation". *Journal of cheminformatics* 6:1, 2014, pp. 1–19.

127. C. *Corbiere*, N. *Thome*, A. *Saporta*, T.-H. *Vu*, M. *Cord*, and P. *Perez*. "Confidence estimation via auxiliary models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

128. T. M. *Dhameliya*, S. J. *Chudasma*, T. M. *Patel*, and B. P. *Dave*. "A review on synthetic account of 1, 2, 4-oxadiazoles as anti-infective agents". *Molecular Diversity*, 2022, pp. 1–14.

129. B. *Ruan*, X. *Tang*, W. *Guo*, Y. *Hu*, and L. *Chen*. "Synthesis and Biological Evaluation of Novel Phthalide Analogs-1, 2, 4-Oxadiazole Hybrids as Potential Anti-Inflammatory Agents". *Chemistry & Biodiversity* 19:8, 2022, e202200039.

130. B.-L. *Sun*, Y.-Y. *Wang*, S. *Yang*, M.-T. *Tu*, Y.-Y. *Shao*, Y. *Hua*, Y. *Zhou*, and C.-X. *Tan*. "Benzamides Substituted with Quinoline-Linked 1, 2, 4-Oxadiazole: Synthesis, Biological Activity and Toxicity to Zebrafish Embryo". *Molecules* 27:12, 2022, p. 3946.

131. Y. M. *Rocha*, E. P. *Magalhães*, M. de *Medeiros Chaves*, M. *Machado Marinho*, V. *Nascimento e Melo de Oliveira*, R. *Nascimento de Oliveira*, T. *Lima Sampaio*, R. R. de *Menezes*, A. *Martins*, and R. *Nicolete*. "Antiparasitary and antiproliferative activities in vitro of a 1, 2, 4-oxadiazole derivative on Trypanosoma cruzi". *Parasitology Research*, 2022, pp. 1–16.

132. I. V. *Tetko*, P. *Karpov*, E. *Bruno*, T. B. *Kimber*, and G. *Godin*. "Augmentation is what you need!" In: *International Conference on Artificial Neural Networks*. Springer. 2019, pp. 831–835.

133. N. L. *Dang*, T. B. *Hughes*, G. P. *Miller*, and S. J. *Swamidass*. "Computational approach to structural alerts: furans, phenols, nitroaromatics, and thiophenes". *Chemical research in toxicology* 30:4, 2017, pp. 1046–1059.

134. R. *Santos*, O. *Ursu*, A. *Gaulton*, A. P. *Bento*, R. S. *Donadi*, C. G. *Bologa*, A. *Karlsson*, B. *Al-Lazikani*, A. *Hersey*, T. I. *Oprea*, et al. "A comprehensive map of molecular drug targets". *Nature reviews Drug discovery* 16:1, 2017, p. 19.

135. R. *Macarron*, M. N. *Banks*, D. *Bojanic*, D. J. *Burns*, D. A. *Cirovic*, T. *Garyantes*, D. V. *Green*, R. P. *Hertzberg*, W. P. *Janzen*, J. W. *Paslay*, et al. "Impact of high-throughput screening in biomedical research". *Nature reviews Drug discovery* 10:3, 2011, pp. 188–195.

136. C. *Parks*, Z. *Gaieb*, and R. E. *Amaro*. "An analysis of proteochemometric and conformal prediction machine learning protein-ligand binding affinity models". *Frontiers in molecular biosciences* 7, 2020, p. 93.

137. E. *Martin*, P. *Mukherjee*, D. *Sullivan*, and J. *Jansen*. "Profile-QSAR: a novel meta-QSAR method that combines activities across the kinase family to accurately predict affinity, selectivity, and cellular activity". *Journal of chemical information and modeling* 51:8, 2011, pp. 1942–1956.

138. E. *Martin* and P. *Mukherjee*. "Kinase-kernel models: accurate in silico screening of 4 million compounds across the entire human kinome". *Journal of chemical information and modeling* 52:1, 2012, pp. 156–170.

139. C. *Fare*, L. *Turcani*, and E. O. *Pyzer-Knapp*. "Powerful, transferable representations for molecules through intelligent task selection in deep multitask networks". *Physical Chemistry Chemical Physics* 22:23, 2020, pp. 13041–13048.

140. R. *Rodrıguez-Pérez* and J. *Bajorath*. "Prediction of compound profiling matrices, part II: relative performance of multitask deep learning and random forest classification on the basis of varying amounts of training data". *ACS omega* 3:9, 2018, pp. 12033–12040.

141. B. *Ramsundar*, B. *Liu*, Z. *Wu*, A. *Verras*, M. *Tudor*, R. P. *Sheridan*, and V. *Pande*. "Is multitask deep learning practical for pharma?" *Journal of chemical information and modeling* 57:8, 2017, pp. 2068–2076.

142. R. *Rodriguez-Perez* and J. *Bajorath*. "Multitask machine learning for classifying highly and weakly potent kinase inhibitors". *Acs Omega* 4:2, 2019, pp. 4367–4375.

143. A. *Mayr*, G. *Klambauer*, T. *Unterthiner*, M. *Steijaert*, J. K. *Wegner*, H. *Ceulemans*, D.-A. *Clevert*, and S. *Hochreiter*. "Large-scale comparison of machine learning methods for drug target prediction on ChEMBL". *Chemical science* 9:24, 2018, pp. 5441–5451.

144. G. J. van *Westen*, J. K. *Wegner*, A. P. *IJzerman*, H. W. van *Vlijmen*, and A. *Bender*. "Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets". *MedChemComm* 2:1, 2011, pp. 16–30.

145. K. *Abbasi*, P. *Razzaghi*, A. *Poso*, S. *Ghanbari-Ara*, and A. *Masoudi-Nejad*. "Deep learning in drug target interaction prediction: current and future perspectives". *Current Medicinal Chemistry* 28:11, 2021, pp. 2100–2113.

146. A. *Cichońska*, B. *Ravikumar*, R. J. *Allaway*, F. *Wan*, S. *Park*, O. *Isayev*, S. *Li*, M. *Mason*, A. *Lamb*, Z. *Tanoli*, et al. "Crowdsourced mapping of unexplored target space of kinase inhibitors". *Nature communications* 12:1, 2021, pp. 1–18.

147. H. *Gaspar*, M. *Ahmed*, T. *Edlich*, B. *Fabian*, Z. *Varszegi*, M. *Segler*, J. *Meyers*, and M. *Fiscato*. "Proteochemometric Models Using Multiple Sequence Alignments and a Subword Segmented Masked Language Model". *ChemRxiv preprint (10.26434/chemrxiv.14604720.v1)*, 2021.

148. K. *Tian*, M. *Shao*, Y. *Wang*, J. *Guan*, and S. *Zhou*. "Boosting compound-protein interaction prediction by deep learning". *Methods* 110, 2016, pp. 64–72.

149. M. *Kuhn*, D. *Szklarczyk*, S. *Pletscher-Frankild*, T. H. *Blicher*, C. *Von Mering*, L. J. *Jensen*, and P. *Bork*. "STITCH 4: integration of protein–chemical interactions with user data". *Nucleic acids research* 42:D1, 2014, pp. D401–D407.

150. L. *Chen*, X. *Tan*, D. *Wang*, F. *Zhong*, X. *Liu*, T. *Yang*, X. *Luo*, K. *Chen*, H. *Jiang*, and M. *Zheng*. "TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments". *Bioinformatics* 36:16, 2020, pp. 4406–4414.

151. J. *Jiménez*, M. *Skalic*, G. *Martinez-Rosell*, and G. *De Fabritiis*. "K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks". *Journal of chemical information and modeling* 58:2, 2018, pp. 287–296.

152. H. *Hassan-Harrirou*, C. *Zhang*, and T. *Lemmin*. "RosENet: improving binding affinity prediction by leveraging molecular mechanics energies with an ensemble of 3D convolutional neural networks". *Journal of chemical information and modeling* 60:6, 2020, pp. 2791–2802.

153. M. *Ragoza*, J. *Hochuli*, E. *Idrobo*, J. *Sunseri*, and D. R. *Koes*. "Protein–ligand scoring with convolutional neural networks". *Journal of chemical information and modeling* 57:4, 2017, pp. 942–957.

154. J. *Lim*, S. *Ryu*, K. *Park*, Y. J. *Choe*, J. *Ham*, and W. Y. *Kim*. "Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation". *Journal of chemical information and modeling* 59:9, 2019, pp. 3981–3988.

155. W. *Torng* and R. B. *Altman*. "Graph convolutional neural networks for predicting drug-target interactions". *Journal of chemical information and modeling* 59:10, 2019, pp. 4131–4149.

156. S. *Zheng*, Y. *Li*, S. *Chen*, J. *Xu*, and Y. *Yang*. "Predicting drug–protein interaction using quasi-visual question answering system". *Nature Machine Intelligence* 2:2, 2020, pp. 134–140.

157. K. Y. *Gao*, A. *Fokoue*, H. *Luo*, A. *Iyengar*, S. *Dey*, and P. *Zhang*. "Interpretable Drug Target Prediction Using Deep Neural Representation." In: *IJCAI*. Vol. 2018. 2018, pp. 3371–3377.

158. M. *Volkov*, J.-A. *Turk*, N. *Drizard*, N. *Martin*, B. *Hoffmann*, Y. *Gaston-Mathé*, and D. *Rognan*. "On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks". *J. Med. Chem.*, 2022.

159. A. *Paszke*, S. *Gross*, F. *Massa*, A. *Lerer*, J. *Bradbury*, G. *Chanan*, T. *Killeen*, Z. *Lin*, N. *Gimelshein*, L. *Antiga*, et al. "Pytorch: An imperative style, high-performance deep learning library". *Advances in neural information processing systems* 32, 2019, pp. 8026–8037.

160. E. *Nazarshodeh*, R. *Sheikhpour*, S. *Gharaghani*, and M. *Sarram*. "A novel proteochemometrics model for predicting the inhibition of nine carbonic anhydrase isoforms based on supervised Laplacian score and k-nearest neighbour regression". *SAR and QSAR in Environmental Research* 29:6, 2018, pp. 419–437.

161. V. I. *Levenshtein*. "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet physics doklady* 10:8, 1966, pp. 707–710.

162. T. T. *Tanimoto*. "Elementary mathematical theory of classification and prediction". *IBM Internal Report*, 1958.

163. J. *Sieg*, F. *Flachsenberg*, and M. *Rarey*. "In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening". *Journal of chemical information and modeling* 59:3, 2019, pp. 947–961.

164. M. K. *Gilson*, T. *Liu*, M. *Baitaluk*, G. *Nicola*, L. *Hwang*, and J. *Chong*. "BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology". *Nucleic acids research* 44:D1, 2016, pp. D1045–D1053.

165. K. *Abbasi*, P. *Razzaghi*, A. *Poso*, M. *Amanlou*, J. B. *Ghasemi*, and A. *Masoudi-Nejad*. "DeepCDA: deep cross-domain compound–protein affinity prediction through LSTM and convolutional neural networks". *Bioinformatics* 36:17, 2020, pp. 4633–4642.

166. L. *Breiman*. "Bagging predictors". *Machine learning* 24:2, 1996, pp. 123–140.

167. Z.-H. *Zhou*. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2019.

168. T. F. *Truong Jr*. "Interpretable deep learning framework for binding affinity prediction". PhD thesis. Massachusetts Institute of Technology, 2020.

169. A. *Stank*, D. B. *Kokh*, J. C. *Fuller*, and R. C. *Wade*. "Protein binding pocket dynamics". *Accounts of chemical research* 49:5, 2016, pp. 809–815.

170. E. *Fischer*. "Einfluss der Configuration auf die Wirkung der Enzyme". *Berichte der deutschen chemischen Gesellschaft* 27:3, 1894, pp. 2985–2993.

171. A. *Vulpetti* and R. *Bosotti*. "Sequence and structural analysis of kinase ATP pocket residues". *Il farmaco* 59:10, 2004, pp. 759–765.

172. P. *Cohen*, D. *Cross*, and P. *Janne*. "Kinase drug discovery 20 years after imatinib: progress and future directions". *Nature Reviews Drug Discovery*, 2021.

173. P. *Cohen*. "Protein kinases—the major drug targets of the twenty-first century?" *Nature reviews Drug discovery* 1:4, 2002, pp. 309–315.

174. P. *Cohen* and D. R. *Alessi*. "Kinase drug discovery–what's next in the field?" *ACS chemical biology* 8:1, 2013, pp. 96–104.

175. S. *Laufer*, K. *Briner*, J. *Bajorath*, G. I. *Georg*, and S. *Wang*. *New Horizons in Drug Discovery-Understanding and Advancing Kinase Inhibitors: Special Issue and Call for Papers*. 2020.

176. A. *Volkamer*, S. *Eid*, S. *Turk*, F. *Rippmann*, and S. *Fulle*. "Identification and visualization of kinase-specific subpockets". *Journal of chemical information and modeling* 56:2, 2016, pp. 335–346.

177. T. *Blaschke*, F. *Miljkovic*, and J. *Bajorath*. "Prediction of different classes of promiscuous and nonpromiscuous compounds using machine learning and nearest neighbor analysis". *ACS Omega* 4:4, 2019, pp. 6883–6890.

178. Y.-C. *Lo*, T. *Liu*, K. M. *Morrissey*, S. *Kakiuchi-Kiyota*, A. R. *Johnson*, F. *Broccatelli*, Y. *Zhong*, A. *Joshi*, and R. B. *Altman*. "Computational analysis of kinase inhibitor selectivity using structural knowledge". *Bioinformatics* 35:2, 2019, pp. 235–242.

179. P. M.-U. *Ung*, R. *Rahman*, and A. *Schlessinger*. "Redefining the protein kinase conformational space with machine learning". *Cell chemical biology* 25:7, 2018, pp. 916–924.

180. L.-C. *Huang*, W. *Yeung*, Y. *Wang*, H. *Cheng*, A. *Venkat*, S. *Li*, P. *Ma*, K. *Rasheed*, and N. *Kannan*. "Quantitative Structure–Mutation–Activity Relationship Tests (QSMART) model for protein kinase inhibitor response prediction". *BMC bioinformatics* 21:1, 2020, pp. 1–22.

181. K. *Koras*, E. *Kizling*, D. *Juraeva*, E. *Staub*, and E. *Szczurek*. "Interpretable deep recommender system model for prediction of kinase inhibitor efficacy across cancer cell lines". *bioRxiv*, 2021.

182. E. J. *Martin*, V. R. *Polyakov*, L. *Tian*, and R. C. *Perez*. "Profile-QSAR 2.0: kinase virtual screening accuracy comparable to four-concentration IC50s for realistically novel compounds". *Journal of chemical information and modeling* 57:8, 2017, pp. 2077–2088.

183. R. P. *Sheridan*, K. *Nam*, V. N. *Maiorov*, D. R. *McMasters*, and W. D. *Cornell*. "QSAR models for predicting the similarity in binding profiles for pairs of protein kinases and the variation of models between experimental data sets". *Journal of chemical information and modeling* 49:8, 2009, pp. 1974–1985.

184. S. K. *Hanks* and T. *Hunter*. "The eukaryotic protein kinase superfamily: kinase (catalytic) domain structure and classification 1". *The FASEB journal* 9:8, 1995, pp. 576–596.

185. V. *Modi* and R. L. *Dunbrack*. "A structurally-validated multiple sequence alignment of 497 human protein kinase domains". *Scientific reports* 9:1, 2019, pp. 1–16.

186. H. *Wang*, J. *Qiu*, H. *Liu*, Y. *Xu*, Y. *Jia*, and Y. *Zhao*. "HKPocket: human kinase pocket database for drug design". *BMC bioinformatics* 20:1, 2019, pp. 1–11.

187. G. *Manning*, D. B. *Whyte*, R. *Martinez*, T. *Hunter*, and S. *Sudarsanam*. "The protein kinase complement of the human genome". *Science* 298:5600, 2002, pp. 1912–1934.

188. A. *Fischer*, A. *Baljuls*, J. *Reinders*, E. *Nekhoroshkova*, C. *Sibilski*, R. *Metz*, S. *Albert*, K. *Rajalingam*, M. *Hekman*, and U. R. *Rapp*. "Regulation of RAF activity by 14-3-3 proteins: RAF kinases associate functionally with both homo-and heterodimeric forms of 14-3-3 proteins". *Journal of Biological Chemistry* 284:5, 2009, pp. 3183–3194.

189. U. *Knippschild*, M. *Krüger*, J. *Richter*, P. *Xu*, B. *García-Reyes*, C. *Peifer*, J. *Halekotte*, V. *Bakulev*, and J. *Bischof*. "The CK1 family: contribution to cellular stress response and its role in carcinogenesis". *Frontiers in oncology* 4, 2014, p. 96.

190. R. *Roskoski Jr*. "Classification of small molecule protein kinase inhibitors based upon the structures of their drug-enzyme complexes". *Pharmacological research* 103, 2016, pp. 26–48.

191. P.-K. *Wu* and J.-I. *Park*. "MEK1/2 inhibitors: molecular activity and resistance mechanisms". In: *Seminars in oncology*. Vol. 42. 6. Elsevier. 2015, pp. 849–862.

192. S. *Henikoff* and J. G. *Henikoff*. "Amino acid substitution matrices from protein blocks." *Proceedings of the National Academy of Sciences* 89:22, 1992, pp. 10915–10919.

193. A. S. *Shaw*, A. P. *Kornev*, J. *Hu*, L. G. *Ahuja*, and S. S. *Taylor*. "Kinases and pseudokinases: lessons from RAF". *Mol. Cell. Biol.* 34, 2014, pp. 1538–1546.

194. E. P. *Reddy* and A. K. *Aggarwal*. "The ins and outs of bcr-abl inhibition". *Genes & cancer* 3:5-6, 2012, pp. 447–454.

195. A. P. *Kornev*, N. M. *Haste*, S. S. *Taylor*, and L. F. *Ten Eyck*. "Surface comparison of active and inactive protein kinases identifies a conserved activation mechanism". *Proc. Natl. Acad. Sci.* 103:47, 2006, pp. 17783–17788.

196. Y. *Cui*, Q. *Dong*, D. *Hong*, and X. *Wang*. "Predicting protein-ligand binding residues with deep convolutional neural networks". *BMC bioinformatics* 20:1, 2019, pp. 1–12.

197. N. *Jaques*, S. *Gu*, D. *Bahdanau*, J. M. *Hernández-Lobato*, R. E. *Turner*, and D. *Eck*. "Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1645–1654.

198. M. *Olivecrona*, T. *Blaschke*, O. *Engkvist*, and H. *Chen*. "Molecular de-novo design through deep reinforcement learning". *Journal of cheminformatics* 9:1, 2017, pp. 1–14.

199. G. L. *Guimaraes*, B. *Sanchez-Lengeling*, C. *Outeiral*, P. L. C. *Farias*, and A. *Aspuru-Guzik*. "Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models". *arXiv preprint arXiv:1705.10843*, 2017.

200. M. *Simonovsky* and N. *Komodakis*. "Graphvae: Towards generation of small graphs using variational autoencoders". In: *International conference on artificial neural networks*. Springer. 2018, pp. 412–422.

201. T. N. *Kipf* and M. *Welling*. "Variational graph auto-encoders". *arXiv preprint arXiv:1611.07308*, 2016.

202. N. *De Cao* and T. *Kipf*. "MolGAN: An implicit generative model for small molecular graphs". *arXiv preprint arXiv:1805.11973*, 2018.

203. R. *Liao*, Y. *Li*, Y. *Song*, S. *Wang*, W. *Hamilton*, D. K. *Duvenaud*, R. *Urtasun*, and R. *Zemel*. "Efficient graph generation with graph recurrent attention networks". *Advances in neural information processing systems* 32, 2019.

204. Q. *Liu*, M. *Allamanis*, M. *Brockschmidt*, and A. *Gaunt*. "Constrained graph variational autoencoders for molecule design". *Advances in neural information processing systems* 31, 2018.

205. D. *Rezende* and S. *Mohamed*. "Variational inference with normalizing flows". In: *International conference on machine learning*. PMLR. 2015, pp. 1530–1538.

206. E. *Bengio*, M. *Jain*, M. *Korablyov*, D. *Precup*, and Y. *Bengio*. "Flow network based generative models for non-iterative diverse candidate generation". *Advances in Neural Information Processing Systems* 34, 2021, pp. 27381–27394.

207. M. *Jain*, E. *Bengio*, A. *Hernandez-Garcia*, J. *Rector-Brooks*, B. F. *Dossou*, C. A. *Ekbote*, J. *Fu*, T. *Zhang*, M. *Kilgour*, D. *Zhang*, et al. "Biological Sequence Design with GFlowNets". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 9786–9801.

208. M. *Xu*, L. *Yu*, Y. *Song*, C. *Shi*, S. *Ermon*, and J. *Tang*. "GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation". In: *The Tenth International Conference on Learning Representations, ICLR*. 2022.

209. E. *Hoogeboom*, V. G. *Satorras*, C. *Vignac*, and M. *Welling*. "Equivariant diffusion for molecule generation in 3d". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 8867–8887.

210. D. P. *Kingma* and M. *Welling*. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014*. 2014.

211. K. *Sohn*, H. *Lee*, and X. *Yan*. "Learning structured output representation using deep conditional generative models". In: *Advances in neural information processing systems*. 2015, pp. 3483–3491.

212. R. J. *Williams*. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". *Machine learning* 8:3-4, 1992, pp. 229–256.

213. N. *Chomsky*. "Three models for the description of language". *IRE Transactions on information theory* 2:3, 1956, pp. 113–124.

214. J. E. *Hopcroft* and J. D. *Ullman*. *Formal Languages and Their Relation to Automata*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1969.

215. A. *Joulin* and T. *Mikolov*. "Inferring algorithmic patterns with stack-augmented recurrent nets". In: *Advances in neural information processing systems*. 2015, pp. 190–198.

216. M. J. *Garnett*, E. J. *Edelman*, S. J. *Heidorn*, C. D. *Greenman*, A. *Dastur*, K. W. *Lau*, P. *Greninger*, I. R. *Thompson*, X. *Luo*, J. *Soares*, et al. "Systematic identification of genomic markers of drug sensitivity in cancer cells". *Nature* 483:7391, 2012, pp. 570–575.

217. A. *Lin*, C. J. *Giuliano*, A. *Palladino*, K. M. *John*, C. *Abramowicz*, M. L. *Yuan*, E. L. *Sausville*, D. A. *Lukow*, L. *Liu*, A. R. *Chait*, et al. "Off-target toxicity is a common mechanism of action of cancer drugs undergoing clinical trials". *Science translational medicine* 11:509, 2019, eaaw8412.

218. B. *Verbist*, G. *Klambauer*, L. *Vervoort*, W. *Talloen*, Z. *Shkedy*, O. *Thas*, A. *Bender*, H. W. *Göhlmann*, S. *Hochreiter*, Q. *Consortium*, et al. "Using transcriptomics to guide lead optimization in drug discovery projects: Lessons learned from the QSTAR project". *Drug discovery today* 20:5, 2015, pp. 505–513.

219. J. *Dopazo*. "Genomics and transcriptomics in drug discovery". *Drug discovery today* 19:2, 2014, pp. 126–132.

220. M. *Wehling*. "Assessing the translatability of drug projects: what needs to be scored to predict success?" *Nature reviews Drug discovery* 8:7, 2009, pp. 541–546.

221. O. *Méndez-Lucio*, B. *Baillif*, D.-A. *Clevert*, D. *Rouquié*, and J. *Wichard*. "De novo generation of hit-like molecules from gene expression signatures using artificial intelligence". *Nature communications* 11:1, 2020, pp. 1–10.

222. R. *Shayakhmetov*, M. *Kuznetsov*, A. *Zhebrak*, A. *Kadurin*, S. *Nikolenko*, A. *Aliper*, and D. *Polykovskiy*. "Molecular generation for desired transcriptome changes with adversarial autoencoders". *Frontiers in pharmacology*, 2020, p. 269.

223. S. *Joo*, M. S. *Kim*, J. *Yang*, and J. *Park*. "Generative Model for Proposing Drug Candidates Satisfying Anticancer Properties Using a Conditional Variational Autoencoder". *ACS omega* 5:30, 2020, pp. 18642–18650.

224. R. H. *Shoemaker*. "The NCI60 human tumour cell line anticancer drug screen". *Nature Reviews Cancer* 6:10, 2006, p. 813.

225. J. N. *Weinstein*, E. A. *Collisson*, G. B. *Mills*, K. R. M. *Shaw*, B. A. *Ozenberger*, K. *Ellrott*, I. *Shmulevich*, C. *Sander*, J. M. *Stuart*, C. G. A. R. *Network*, et al. "The cancer genome atlas pan-cancer analysis project". *Nature genetics* 45:10, 2013, p. 1113.

226. A. *Oskooei*, M. *Manica*, R. *Mathis*, and M. R. *Martınez*. "Network-based biased tree ensembles (NetBiTE) for drug sensitivity prediction and drug sensitivity biomarker identification in cancer". *Scientific reports* 9:1, 2019, pp. 1–13.

227. W. *Yang*, J. *Soares*, P. *Greninger*, *Edelman*, et al. "Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells". *Nucleic acids research* 41:D1, 2012, pp. D955–D961.

228. A. *Gaulton*, A. *Hersey*, M. *Nowotka*, A. P. *Bento*, J. *Chambers*, D. *Mendez*, P. *Mutowo*, F. *Atkinson*, L. J. *Bellis*, E. *Cibrián-Uhalte*, et al. "The ChEMBL database in 2017". *Nucleic acids research* 45:D1, 2016, pp. D945–D954.

229. R. J. *Williams* and D. *Zipser*. "A learning algorithm for continually running fully recurrent neural networks". *Neural computation* 1:2, 1989, pp. 270–280.

230. S. R. *Bowman*, L. *Vilnis*, O. *Vinyals*, A. *Dai*, R. *Jozefowicz*, and S. *Bengio*. "Generating Sentences from a Continuous Space". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2016, pp. 10–21. DOI: 10.18653/v1/K16-1002.

231. M. *Manica*, A. *Oskooei*, J. *Born*, V. *Subramanian*, J. *Sáez-Rodrıguez*, and M. *Rodríguez Martínez*. "Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders". *Molecular pharmaceutics* 16:12, 2019, pp. 4797–4806.

232. J. *Barretina*, G. *Caponigro*, N. *Stransky*, K. *Venkatesan*, et al. "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity". *Nature* 483:7391, 2012, p. 603.

233. M. *Ghandi*, F. W. *Huang*, J. *Jané-Valbuena*, G. V. *Kryukov*, C. C. *Lo*, E. R. *McDonald*, J. *Barretina*, E. T. *Gelfand*, C. M. *Bielski*, H. *Li*, et al. "Next-generation characterization of the Cancer Cell Line Encyclopedia". *Nature* 569:7757, 2019, p. 503.

234. P. *Shivakumar* and M. *Krauthammer*. "Structural similarity assessment for drug sensitivity prediction in cancer". In: *BMC bioinformatics*. Vol. 10. Springer. 2009, S17.

235. J. *Lao*, J. *Madani*, T. *Puértolas*, M. *Álvarez*, A. *Hernández*, R. *Pazo-Cid*, Á. *Artal*, and A. *Antón Torres*. "Liposomal doxorubicin in the treatment of breast cancer patients: a review". *Journal of drug delivery* 2013, 2013.

236. R. *Poojari*. "Embelin–a drug of antiquity: shifting the paradigm towards modern medicine". *Expert opinion on investigational drugs* 23:3, 2014, pp. 427–444.

237. Y.-J. *Cheng*, H.-S. *Jiang*, S.-L. *Hsu*, L.-C. *Lin*, C.-L. *Wu*, V. K. *Ghanta*, and C.-M. *Hsueh*. "XIAP-mediated protection of H460 lung cancer cells against cisplatin". *European journal of pharmacology* 627:1-3, 2010, pp. 75–84.

238. X. *Zhang*, W. *Wang*, L. D. *True*, R. L. *Vessella*, and T. K. *Takayama*. "Protease-activated receptor-1 is upregulated in reactive stroma of primary prostate cancer and bone metastasis". *The Prostate* 69:7, 2009, pp. 727–736. DOI: 10.1002/pros.20920.

239. M. *Gorska*, A. *Kuban-Jankowska*, R. *Milczarek*, and M. *Wozniak*. "Nitro-oxidative Stress Is Involved in Anticancer Activity of 17β-Estradiol Derivative in Neuroblastoma Cells". *Anticancer research* 36, 2016, pp. 1693–8.

240. D. W. *Gammon*, D. J. *Steenkamp*, V. *Mavumengwana*, M. J. *Marakalala*, T. T. *Mudzunga*, R. *Hunter*, and M. *Munyololo*. "Conjugates of plumbagin and phenyl-2-amino-1-thioglucoside inhibit MshB, a deacetylase involved in the biosynthesis of mycothiol". *Bioorganic & medicinal chemistry* 18:7, 2010, pp. 2501–2514.

241. X. *Zhang*, C. *Yang*, X. *Rao*, and J. *Xiong*. "Plumbagin shows anti-cancer activity in human breast cancer cells by the upregulation of p53 and p21 and suppression of G1 cell cycle regulators." *European journal of gynaecological oncology* 37:1, 2016, pp. 30–35.

242. A. *Kawiak*, A. *Domachowska*, A. *Jaworska*, and E. *Lojkowska*. "Plumbagin sensitizes breast cancer cells to tamoxifen-induced cell death through GRP78 inhibition and Bik upregulation". *Scientific reports* 7, 2017, p. 43781.

243. P. *Dandawate*, A. *Ahmad*, J. *Deshpande*, K. V. *Swamy*, E. M. *Khan*, M. *Khetmalas*, S. *Padhye*, and F. *Sarkar*. "Anticancer phytochemical analogs 37: synthesis, characterization, molecular docking and cytotoxicity of novel plumbagin hydrazones against breast cancer cells". *Bioorganic & medicinal chemistry letters* 24:13, 2014, pp. 2900–2904.

244. L. *Girsh*. *Lipid-containing compositions and methods of using them*. US Patent App. 11/501,380. 2007.

245. P. *Kaur*, T. *Garg*, G. *Rath*, R. *Murthy*, and A. K. *Goyal*. "Surfactant-based drug delivery systems for treating drug-resistant lung cancer". *Drug delivery* 23:3, 2016, pp. 717–728.

246. J. *Klaveness*, P. *Rongved*, A. *Høgset*, H. *Tolleshaug*, A. *Cuthbertson*, A. *Godal*, L. *Hoff*, G. *Gogstad*, K. *Bryn*, A. *Naevestad*, et al. *Diagnostic/therapeutic agents*. US Patent 6,680,047. 2004.

247. B. *Schölkopf*, A. *Smola*, and K.-R. *Müller*. "Nonlinear component analysis as a kernel eigenvalue problem". *Neural computation* 10:5, 1998, pp. 1299–1319.

248. G. R. *Bickerton*, G. V. *Paolini*, J. *Besnard*, S. *Muresan*, and A. L. *Hopkins*. "Quantifying the chemical beauty of drugs". *Nature chemistry* 4:2, 2012, p. 90.

249. J. S. *Delaney*. "ESOL: Estimating Aqueous Solubility Directly from Molecular Structure". *Journal of Chemical Information and Computer Sciences* 44:3, 2004, pp. 1000–1005. DOI: 10.1021/ci034243x.

250. C. W. *Coley*, L. *Rogers*, W. H. *Green*, and K. F. *Jensen*. "SCScore: synthetic complexity learned from a reaction corpus". *Journal of chemical information and modeling* 58:2, 2018, pp. 252–261.

251. K. T. *Savjani*, A. K. *Gajjar*, and J. K. *Savjani*. "Drug solubility: importance and enhancement techniques". *ISRN pharmaceutics* 2012, 2012.

252. M. C. *Sorkun*, A. *Khetan*, and S. *Er*. "AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds". *Scientific data* 6:1, 2019, pp. 1–8.

253. D. S. *Wishart*, Y. D. *Feunang*, A. C. *Guo*, E. J. *Lo*, A. *Marcu*, J. R. *Grant*, T. *Sajed*, D. *Johnson*, C. *Li*, Z. *Sayeeda*, et al. "DrugBank 5.0: a major update to the DrugBank database for 2018". *Nucleic acids research* 46:D1, 2018, pp. D1074–D1082.

254. T. *Yu*, S. *Kumar*, A. *Gupta*, S. *Levine*, K. *Hausman*, and C. *Finn*. "Gradient Surgery for Multi-Task Learning". *Advances in Neural Information Processing Systems* 33, 2020.

255. F. *Häse*, L. M. *Roch*, and A. *Aspuru-Guzik*. "Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories". *Chemical science* 9:39, 2018, pp. 7642–7655.

256. H. *Sharifi-Noghabi*, S. *Peng*, O. *Zolotareva*, C. C. *Collins*, and M. *Ester*. "AITL: Adversarial Inductive Transfer Learning with input and output space adaptation for pharmacogenomics". *Bioinformatics* 36:Supplement_1, 2020, pp. i380–i388.

257. C. *Drosten* et al. "Identification of a novel coronavirus in patients with severe acute respiratory syndrome". *New England journal of medicine* 348:20, 2003, pp. 1967–1976.

258. A. *Shamshirian*, A. *Hessami*, K. *Heydari*, R. *Alizadeh-Navaei*, M. A. *Ebrahimzadeh*, G. W. *Yip*, R. *Ghasemian*, M. *Sedaghat*, H. *Baradaran*, S. M. *Yazdi*, et al. "The role of hydroxychloroquine in COVID-19 treatment: a systematic review and meta-analysis". *Ann Acad Med Singap* 49, 2020, pp. 789–800.

259. Y. N. *Lamb*. "Remdesivir: first approval". *Drugs*, 2020, pp. 1–9.

260. R. *Beckerman*, A. *Gori*, S. *Jeyakumar*, J. J. *Malin*, R. *Paredes*, P. *Povoa*, N. J. *Smith*, and A. *Teixeira-Pinto*. "Remdesivir for the treatment of patients hospitalized with COVID-19 receiving supplemental oxygen: a targeted literature review and meta-analysis". *Scientific Reports* 12:1, 2022, pp. 1–11.

261. A. *Zhavoronkov* et al. "Potential non-covalent SARS-CoV-2 3C-like protease inhibitors designed using generative deep learning approaches and reviewed by human medicinal chemist in virtual reality". *ChemRxiv*, 2020.

262. B. *Tang*, F. *He*, D. *Liu*, M. *Fang*, Z. *Wu*, and D. *Xu*. "AI-aided design of novel targeted covalent inhibitors against SARS-CoV-2". *bioRxiv*, 2020.

263. N. *Bung*, S. R. *Krishnan*, G. *Bulusu*, and A. *Roy*. "De novo design of new chemical entities for SARS-CoV-2 using artificial intelligence". *Future medicinal chemistry* 0, 2021.

264. T. *Aumentado-Armstrong*. "Latent molecular optimization for targeted therapeutic design". *arXiv preprint arXiv:1809.02032*, 2018.

265. V. *Chenthamarakshan*, P. *Das*, S. *Hoffman*, H. *Strobelt*, I. *Padhi*, K. W. *Lim*, B. *Hoover*, M. *Manica*, J. *Born*, T. *Laino*, et al. "Cogmol: Target-specific and selective drug design for covid-19 using deep generative models". *Advances in Neural Information Processing Systems* 33, 2020.

266. M. *Skalic*, D. *Sabbadin*, B. *Sattarov*, S. *Sciabola*, and G. *De Fabritiis*. "From target to drug: Generative modeling for the multimodal structure-based ligand design". *Molecular pharmaceutics* 16:10, 2019, pp. 4282–4291.

267. S. R. *Krishnan*, N. *Bung*, G. *Bulusu*, and A. *Roy*. "Accelerating De Novo Drug Design against Novel Proteins Using Deep Learning". *Journal of Chemical Information and Modeling*, 2021.

268. D. *Grechishnikova*. "Transformer neural network for protein-specific de novo drug generation as a machine translation problem". *Scientific reports* 11:1, 2021, pp. 1–13.

269. S. *Luo*, J. *Guan*, J. *Ma*, and J. *Peng*. "A 3D generative model for structure-based drug design". *Advances in Neural Information Processing Systems* 34, 2021, pp. 6229–6239.

270. M. *Ragoza*, T. *Masuda*, and D. R. *Koes*. "Generating 3D molecules conditional on receptor binding sites with deep generative models". *Chemical science* 13:9, 2022, pp. 2701–2713.

271. U. *Consortium*. "UniProt: a worldwide hub of protein knowledge". *Nucleic acids research* 47:D1, 2019, pp. D506–D515.

272. S. *El-Gebali*, J. *Mistry*, A. *Bateman*, S. R. *Eddy*, A. *Luciani*, S. C. *Potter*, M. *Qureshi*, L. J. *Richardson*, G. A. *Salazar*, A. *Smart*, et al. "The Pfam protein families database in 2019". *Nucleic acids research* 47:D1, 2019, pp. D427–D432.

273. T. *Kalliokoski*, C. *Kramer*, A. *Vulpetti*, and P. *Gedeck*. "Comparability of mixed IC 50 data–a statistical analysis". *PloS one* 8:4, 2013, e61007.

274. A. *Gonczarek*, J. M. *Tomczak*, S. *Zaręba*, J. *Kaczmar*, P. *Dąbrowski*, and M. J. *Walczak*. "Interaction prediction in structure-based virtual screening using deep learning". *Computers in biology and medicine* 100, 2018, pp. 253–258.

275. M. K. *Gilson*. "An Introduction to Protein-Ligand Binding for BindingDB Users". Dataset overview distributed by the BindingDB authors. 2010.

276. X. *Du*, Y. *Li*, Y.-L. *Xia*, S.-M. *Ai*, J. *Liang*, P. *Sang*, X.-L. *Ji*, and S.-Q. *Liu*. "Insights into protein–ligand interactions: mechanisms, models, and methods". *International journal of molecular sciences* 17:2, 2016, p. 144.

277. J. *Arús-Pous*, T. *Blaschke*, S. *Ulander*, J.-L. *Reymond*, H. *Chen*, and O. *Engkvist*. "Exploring the GDB-13 chemical space using deep generative models". *Journal of cheminformatics* 11:1, 2019, pp. 1–14.

278. D. E. *Gordon*, G. M. *Jang*, M. *Bouhaddou*, J. *Xu*, K. *Obernier*, K. M. *White*, M. J. *O'Meara*, V. V. *Rezelj*, J. Z. *Guo*, D. L. *Swaney*, et al. "A SARS-CoV-2 protein interaction map reveals targets for drug repurposing". *Nature*, 2020, pp. 1–13.

279. *Pubchem Compound Database. (2S)-4-Phenyl-3-buten-2-ol*. URL: https://pubchem.ncbi.nlm.nih.gov/compound/10975641.

280. L. *McInnes*, J. *Healy*, and J. *Melville*. "Umap: Uniform manifold approximation and projection for dimension reduction". *arXiv preprint arXiv:1802.03426*, 2018.

281. D. *Probst* and J.-L. *Reymond*. "FUn: a framework for interactive visualizations of large, high-dimensional datasets on the web". *Bioinformatics* 34:8, 2018, pp. 1433–1435.

282. A. *Peón*, S. *Naulaerts*, and P. J. *Ballester*. "Predicting the reliability of drug-target interaction predictions with maximum coverage of target space". *Scientific reports* 7:1, 2017, pp. 1–11.

283. F. *Miljković* and J. *Bajorath*. "Data-driven exploration of selectivity and off-target activities of designated chemical probes". *Molecules* 23:10, 2018, p. 2434.

284. L. *Chan*, R. *Kumar*, M. *Verdonk*, and C. *Poelking*. "3D pride without 2D prejudice: Bias-controlled multi-level generative models for structure-based ligand design". *arXiv preprint arXiv:2204.10663*, 2022.

285. W. *Gao* and C. W. *Coley*. "The synthesizability of molecules proposed by generative models". *Journal of chemical information and modeling* 60:12, 2020, pp. 5714–5723.

286. S.-r. *Li*, Z.-j. *Tang*, Z.-h. *Li*, and X. *Liu*. "Searching therapeutic strategy of new coronavirus pneumonia from angiotensin-converting enzyme 2: the target of COVID-19 and SARS-CoV". *European Journal of Clinical Microbiology & Infectious Diseases* 39:6, 2020, p. 1021.

287. H. *Zhang*, J. M. *Penninger*, Y. *Li*, N. *Zhong*, and A. S. *Slutsky*. "Angiotensin-converting enzyme 2 (ACE2) as a SARS-CoV-2 receptor: molecular mechanisms and potential therapeutic target". *Intensive care medicine* 46:4, 2020, pp. 586–590.

288. D. L. *McKee*, A. *Sternberg*, U. *Stange*, S. *Laufer*, and C. *Naujokat*. "Candidate drugs against SARS-CoV-2 and COVID-19". *Pharmacological Research*, 2020, p. 104859.

289. K. *Teralı*, B. *Baddal*, and H. O. *Gülcan*. "Prioritizing potential ACE2 inhibitors in the COVID-19 pandemic: insights from a molecular mechanics-assisted structure-based virtual screening experiment". *Journal of Molecular Graphics and Modelling*, 2020, p. 107697.

290. S. *Ray*, S. *Lall*, A. *Mukhopadhyay*, S. *Bandyopadhyay*, and A. *Schönhuth*. "Predicting potential drug targets and repurposable drugs for COVID-19 via a deep generative model for graphs". *arXiv preprint arXiv:2007.02338*, 2020.

291. I. *Hamming*, W. *Timens*, M. *Bulthuis*, A. *Lely*, G. v. *Navis*, and H. van *Goor*. "Tissue distribution of ACE2 protein, the functional receptor for SARS coronavirus. A first step in understanding SARS pathogenesis". *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland* 203:2, 2004, pp. 631–637.

292. M. *Donoghue*, F. *Hsieh*, E. *Baronas*, K. *Godbout*, M. *Gosselin*, N. *Stagliano*, M. *Donovan*, B. *Woolf*, K. *Robison*, R. *Jeyaseelan*, et al. "A novel angiotensin-converting enzyme–related carboxypeptidase (ACE2) converts angiotensin I to angiotensin 1-9". *Circulation research* 87:5, 2000, e1–e9.

293. S. R. *Tipnis*, N. M. *Hooper*, R. *Hyde*, E. *Karran*, G. *Christie*, and A. J. *Turner*. "A human homolog of angiotensin-converting enzyme cloning and functional expression as a captopril-insensitive carboxypeptidase". *Journal of Biological Chemistry* 275:43, 2000, pp. 33238–33243.

294. G. Y. *Oudit*, M. A. *Crackower*, P. H. *Backx*, and J. M. *Penninger*. "The role of ACE2 in cardiovascular physiology". *Trends in cardiovascular medicine* 13:3, 2003, pp. 93–101.

295. M. A. *Crackower*, R. *Sarao*, G. Y. *Oudit*, C. *Yagil*, I. *Kozieradzki*, S. E. *Scanga*, A. J. *Oliveira-dos-Santos*, J. da *Costa*, L. *Zhang*, Y. *Pei*, et al. "Angiotensin-converting enzyme 2 is an essential regulator of heart function". *Nature* 417:6891, 2002, pp. 822–828.

296. Y. le *Tran* and C. *Forster*. "Angiotensin-(1-7) and the rat aorta: modulation by the endothelium". *Journal of cardiovascular pharmacology* 30:5, 1997, pp. 676–682.

297. C. *Schindler*, P. *Bramlage*, W. *Kirch*, and C. M. *Ferrario*. "Role of the vasodilator peptide angiotensin-(1–7) in cardiovascular drug therapy". *Vascular health and risk management* 3:1, 2007, p. 125.

298. X. *Li*, M. *Molina-Molina*, A. *Abdul-Hafez*, V. *Uhal*, A. *Xaubet*, and B. D. *Uhal*. "Angiotensin converting enzyme-2 is protective but downregulated in human and experimental lung fibrosis". *American Journal of Physiology-Lung Cellular and Molecular Physiology* 295:1, 2008, pp. L178–L185.

299. W. *Li*, M. J. *Moore*, N. *Vasilieva*, J. *Sui*, S. K. *Wong*, M. A. *Berne*, M. *Somasundaran*, J. L. *Sullivan*, K. *Luzuriaga*, T. C. *Greenough*, et al. "Angiotensin-converting enzyme 2 is a functional receptor for the SARS coronavirus". *Nature* 426:6965, 2003, pp. 450–454.

300. F. *Li*, W. *Li*, M. *Farzan*, and S. C. *Harrison*. "Structure of SARS coronavirus spike receptor-binding domain complexed with receptor". *Science* 309:5742, 2005, pp. 1864–1868.

301. M. *Hoffmann*, H. *Kleine-Weber*, S. *Schroeder*, N. *Krüger*, T. *Herrler*, S. *Erichsen*, T. S. *Schiergens*, G. *Herrler*, N.-H. *Wu*, A. *Nitsche*, et al. "SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor". *Cell*, 2020.

302. P. *Zhou*, X.-L. *Yang*, X.-G. *Wang*, B. *Hu*, L. *Zhang*, W. *Zhang*, H.-R. *Si*, Y. *Zhu*, B. *Li*, C.-L. *Huang*, et al. "A pneumonia outbreak associated with a new coronavirus of probable bat origin". *Nature*, 2020, pp. 1–4.

303. W. *Tai*, L. *He*, X. *Zhang*, J. *Pu*, D. *Voronin*, S. *Jiang*, Y. *Zhou*, and L. *Du*. "Characterization of the receptor-binding domain (RBD) of 2019 novel coronavirus: implication for development of RBD protein as a viral attachment inhibitor and vaccine". *Cellular & Molecular Immunology*, 2020, pp. 1–8.

304. W.-H. *Chen*, P. J. *Hotez*, and M. E. *Bottazzi*. "Potential for developing a SARS-CoV receptor-binding domain (RBD) recombinant protein as a heterologous human vaccine against coronavirus infectious disease (COVID)-19". *Human Vaccines & Immunotherapeutics*, 2020, pp. 1–4.

305. Y. *Cao*, T. *Jiang*, and T. *Girke*. "A maximum common substructure-based algorithm for searching and predicting drug-like compounds". *Bioinformatics* 24:13, 2008, pp. i366–i374.

306. N. Y. *Pshenichnaya*, V. *Bulgakova*, N. *Lvov*, A. *Poromov*, E. *Selkova*, A. *Grekova*, I. *Shestakova*, V. *Maleev*, and I. *Leneva*. "Clinical efficacy of umifenovir in influenza and ARVI (study ARBITR)". *Therapeutic archive* 91:3, 2019, pp. 56–63.

307. Y. S. *Boriskin*, E.-I. *Pécheur*, and S. J. *Polyak*. "Arbidol: a broad-spectrum antiviral that inhibits acute and chronic HCV infection". *Virology journal* 3:1, 2006, p. 56.

308. C. *Liu*, Q. *Zhou*, Y. *Li*, L. V. *Garner*, S. P. *Watkins*, L. J. *Carter*, J. *Smoot*, A. C. *Gregg*, A. D. *Daniels*, S. *Jervey*, et al. "Research and Development on Therapeutic Agents and Vaccines for COVID-19 and Related Human Coronavirus Diseases". *ACS central science* 6:3, 2020, pp. 315–331.

309. S. *Choudhary* and O. *Silakari*. "Scaffold morphing of arbidol (umifenovir) in search of multi-targeting therapy halting the interaction of SARS-CoV-2 with ACE2 and other proteases involved in COVID-19". *Virus research* 289, 2020, p. 198146.

310. A. *Padhi*, A. *Seal*, and T. *Tripathi*. *How does arbidol inhibit the novel coronavirus SARS-CoV-2? Atomistic Insights from Molecular Dynamics Simulations*. 2020.

311. H. *Zhao*, H. *Lam*, X. *Zhou*, K. *To*, J. *Chan*, A. C. *Lee*, J.-P. *Cai*, C. *Chan*, M. L. *Yeung*, A. J. *Zhang*, et al. "Cross-linking peptide and repurposed drugs inhibit both entry pathways of SARS-CoV-2". Under Review in Nature Research. 2020.

312. R. *Khamitov*, S. *Loginova*, V. *Shchukina*, S. *Borisevich*, V. *Maksimov*, and A. *Shuster*. "Antiviral activity of arbidol and its derivatives against the pathogen of severe acute respiratory syndrome in the cell cultures". *Voprosy virusologii* 53:4, 2008, pp. 9–13.

313. Z. *Wang*, B. *Yang*, Q. *Li*, L. *Wen*, and R. *Zhang*. "Clinical features of 69 cases with coronavirus disease 2019 in Wuhan, China". *Clinical infectious diseases*, 2020.

314. S. *Wei*, S. *Xu*, and Y.-H. *Pan*. "Efficacy of arbidol in COVID-19 patients: A retrospective study". *World Journal of Clinical Cases* 9:25, 2021, p. 7350.

315. A. *Di Mola*, A. *Peduto*, A. *La Gatta*, L. *Delang*, B. *Pastorino*, J. *Neyts*, P. *Leyssen*, M. de *Rosa*, and R. *Filosa*. "Structure–activity relationship study of arbidol derivatives as inhibitors of chikungunya virus replication". *Bioorganic & medicinal chemistry* 22:21, 2014, pp. 6014–6025.

316. C. A. *Lipinski*, F. *Lombardo*, B. W. *Dominy*, and P. J. *Feeney*. "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings". *Advanced Drug Delivery Reviews* 23:1, 1997. In Vitro Models for Selection of Development Candidates, pp. 3–25. ISSN: 0169-409X. DOI: https://doi.org/10.1016/S0169-409X(96)00423-1. URL: http://www.sciencedirect.com/science/article/pii/S0169409X96004231.

317. K. *Heiser* et al. "Identification of potential treatments for COVID-19 through artificial intelligence-enabled phenomic analysis of human cells infected with SARS-CoV-2". *bioRxiv*, 2020.

318. J. *Lan*, J. *Ge*, J. *Yu*, S. *Shan*, H. *Zhou*, S. *Fan*, Q. *Zhang*, X. *Shi*, Q. *Wang*, L. *Zhang*, et al. "Structure of the SARS-CoV-2 spike receptor-binding domain bound to the ACE2 receptor". *Nature* 581:7807, 2020, pp. 215–220.

319. N. *Janakarajan*, J. *Born*, and M. *Manica*. "A Fully Differentiable Set Autoencoder". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 3061–3071.

320. R.-R. *Griffiths* and J. M. *Hernández-Lobato*. "Constrained Bayesian optimization for automatic chemical design using variational autoencoders". *Chemical science* 11:2, 2020, pp. 577–586.

321. A. *Gaulton*, A. *Hersey*, M. *Nowotka*, A. P. *Bento*, J. *Chambers*, D. *Mendez*, P. *Mutowo*, F. *Atkinson*, L. J. *Bellis*, E. *Cibrián-Uhalte*, et al. "The ChEMBL database in 2017". *Nucleic acids research* 45:D1, 2017, pp. D945–D954.

322. D. R. *Jones*, M. *Schonlau*, and W. J. *Welch*. "Efficient global optimization of expensive black-box functions". *Journal of Global optimization* 13:4, 1998, pp. 455–492.

323. A. S. *Dhillon*, S. *Hagan*, O. *Rath*, and W. *Kolch*. "MAP kinase signalling pathways in cancer". *Oncogene* 26:22, 2007, pp. 3279–3290.

324. R. *Winter*, F. *Montanari*, A. *Steffen*, H. *Briem*, F. *Noé*, and D.-A. *Clevert*. "Efficient multi-objective molecular optimization in a continuous latent space". *Chemical science* 10:34, 2019, pp. 8016–8024.

325. P. *Schwaller*, D. *Probst*, A. C. *Vaucher*, V. H. *Nair*, D. *Kreutter*, T. *Laino*, and J.-L. *Reymond*. "Mapping the space of chemical reactions using attention-based neural networks". *Nature Machine Intelligence* 3:2, 2021, pp. 144–152.

326. A. *Rives*, J. *Meier*, T. *Sercu*, S. *Goyal*, Z. *Lin*, J. *Liu*, D. *Guo*, M. *Ott*, C. L. *Zitnick*, J. *Ma*, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". *Proceedings of the National Academy of Sciences* 118:15, 2021.

327. J. *Jumper*, R. *Evans*, A. *Pritzel*, T. *Green*, M. *Figurnov*, O. *Ronneberger*, K. *Tunyasuvunakool*, R. *Bates*, A. *Zidek*, A. *Potapenko*, et al. "Highly accurate protein structure prediction with AlphaFold". *Nature*, 2021, pp. 1–11.

328. A. *Krizhevsky*, I. *Sutskever*, and G. E. *Hinton*. "Imagenet classification with deep convolutional neural networks". *Advances in neural information processing systems* 25, 2012, pp. 1097–1105.

329. M.-T. *Luong*, H. *Pham*, and C. D. *Manning*. "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1412–1421.

330. P. *Ramachandran*, N. *Parmar*, A. *Vaswani*, I. *Bello*, A. *Levskaya*, and J. *Shlens*. "Stand-Alone Self-Attention in Vision Models". *Advances in Neural Information Processing Systems* 32, 2019.

331. K. *Lu*, A. *Grover*, P. *Abbeel*, and I. *Mordatch*. "Frozen Pretrained Transformers as Universal Computation Engines". *Proceedings of the AAAI Conference on Artificial Intelligence* 36:7, 2022, pp. 7628–7636.

332. L. *Chen*, K. *Lu*, A. *Rajeswaran*, K. *Lee*, A. *Grover*, M. *Laskin*, P. *Abbeel*, A. *Srinivas*, and I. *Mordatch*. "Decision transformer: Reinforcement learning via sequence modeling". *Advances in neural information processing systems* 34, 2021, pp. 15084–15097.

333. R. *Rao*, N. *Bhattacharya*, N. *Thomas*, Y. *Duan*, X. *Chen*, J. *Canny*, P. *Abbeel*, and Y. S. *Song*. "Evaluating protein transfer learning with TAPE". In: *Advances in Neural Information Processing Systems*. 2019, pp. 9686–9698.

334. J. *Devlin*, M.-W. *Chang*, K. *Lee*, and K. *Toutanova*. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, 2019, pp. 4171–4186.

335. J. *Lu* and Y. *Zhang*. "Unified Deep Learning Model for Multitask Reaction Predictions with Explanation". *Journal of Chemical Information and Modeling* 62:6, 2022, pp. 1376–1387.

336. Z. *Yang*, Z. *Dai*, Y. *Yang*, J. *Carbonell*, R. R. *Salakhutdinov*, and Q. V. *Le*. "Xlnet: Generalized autoregressive pretraining for language understanding". *Advances in neural information processing systems* 32, 2019.

337. D. C. *Elton*, Z. *Boukouvalas*, M. D. *Fuge*, and P. W. *Chung*. "Deep learning for molecular design—a review of the state of the art". *Molecular Systems Design & Engineering* 4:4, 2019, pp. 828–849.

338. Z. *Chen*, M. R. *Min*, S. *Parthasarathy*, and X. *Ning*. "A deep generative model for molecule optimization via one fragment modification". *Nature Machine Intelligence* 3:12, 2021, pp. 1040–1049.

339. Z. *Wu*, K. E. *Johnston*, F. H. *Arnold*, and K. K. *Yang*. "Protein sequence design with deep generative models". *Current Opinion in Chemical Biology* 65, 2021, pp. 18–27.

340. A. *Madani*, B. *McCann*, N. *Naik*, N. S. *Keskar*, N. *Anand*, R. R. *Eguchi*, P.-S. *Huang*, and R. *Socher*. "Progen: Language modeling for protein generation". *NeurIPS 2020 workshop on Machine Learning for Structural Biology (arXiv preprint arXiv:2004.03497)*, 2020.

341. S. *Wang*, Y. *Guo*, Y. *Wang*, H. *Sun*, and J. *Huang*. "SMILES-BERT: large scale unsupervised pre-training for molecular property prediction". In: *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*. 2019, pp. 429–436.

342. H. *Kim*, J. *Lee*, S. *Ahn*, and J. R. *Lee*. "A merged molecular representation learning for molecular properties prediction with a web-based service". *Scientific Reports* 11:1, 2021, pp. 1–9.

343. R. *Irwin*, S. *Dimitriadis*, J. *He*, and E. J. *Bjerrum*. "Chemformer: A Pre-Trained Transformer for Computational Chemistry". *Machine Learning: Science and Technology*, 2021.

344. O. *Mahmood*, E. *Mansimov*, R. *Bonneau*, and K. *Cho*. "Masked graph modeling for molecule generation". *Nature communications* 12:1, 2021, pp. 1–12.

345. J. *Born*, M. *Manica*, J. *Cadow*, G. *Markert*, N. A. *Mill*, M. *Filipavicius*, N. *Janakarajan*, A. *Cardinale*, T. *Laino*, and M. *Rodriguez Martinez*. "Data-driven molecular design for discovery and synthesis of novel ligands: a case study on SARS-CoV-2". *Machine Learning: Science and Technology* 2:2, 2021, p. 025024. ISSN: 2632-2153. DOI: 10.1088/2632-2153/abe808.

346. K. *Maziarz*, H. R. *Jackson-Flux*, P. *Cameron*, F. *Sirockin*, N. *Schneider*, N. *Stiefl*, M. H. S. *Segler*, and M. *Brockschmidt*. "Learning to Extend Molecular Scaffolds with Structural Motifs". In: *The Tenth International Conference on Learning Representations, ICLR*. 2022.

347. J. *Arús-Pous*, A. *Patronov*, E. J. *Bjerrum*, C. *Tyrchan*, J.-L. *Reymond*, H. *Chen*, and O. *Engkvist*. "SMILES-based deep generative scaffold decorator for de-novo drug design". *Journal of Cheminformatics* 12:1, 2020, pp. 1–18.

348. J. *Lim*, S.-Y. *Hwang*, S. *Moon*, S. *Kim*, and W. Y. *Kim*. "Scaffold-based molecular design with a graph generative model". *Chemical science* 11:4, 2020, pp. 1153–1164.

349. Y. *Li*, J. *Hu*, Y. *Wang*, J. *Zhou*, L. *Zhang*, and Z. *Liu*. "Deepscaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning". *Journal of chemical information and modeling* 60:1, 2019, pp. 77–91.

350. Y. *Li*, L. *Zhang*, and Z. *Liu*. "Multi-objective de novo drug design with conditional graph generative model". *Journal of cheminformatics* 10:1, 2018, pp. 1–24.

351. S. *Harel* and K. *Radinsky*. "Prototype-based compound discovery using deep generative models". *Molecular pharmaceutics* 15:10, 2018, pp. 4406–4416.

352. Z. *Zhou*, S. *Kearnes*, L. *Li*, R. N. *Zare*, and P. *Riley*. "Optimization of molecules via deep reinforcement learning". *Scientific reports* 9:1, 2019, pp. 1–10.

353. Ł. *Maziarka*, A. *Pocha*, J. *Kaczmarczyk*, K. *Rataj*, T. *Danel*, and M. *Warchoł*. "Mol-CycleGAN: a generative model for molecular optimization". *Journal of Cheminformatics* 12:1, 2020, pp. 1–18.

354. K. *Song*, X. *Tan*, T. *Qin*, J. *Lu*, and T. *Liu*. "MPNet: Masked and Permuted Pre-training for Language Understanding". In: *Advances in Neural Information Processing Systems 33*. 2020.

355. A. *Van Oord*, N. *Kalchbrenner*, and K. *Kavukcuoglu*. "Pixel recurrent neural networks". In: *International conference on machine learning*. PMLR. 2016, pp. 1747–1756.

356. T. *Wolf*, L. *Debut*, V. *Sanh*, J. *Chaumond*, C. *Delangue*, A. *Moi*, P. *Cistac*, T. *Rault*, R. *Louf*, M. *Funtowicz*, et al. "Transformers: State-of-the-art natural language processing". In: *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 2020, pp. 38–45.

357. D. *Mendez*, A. *Gaulton*, A. P. *Bento*, J. *Chambers*, M. *De Veij*, E. *Félix*, M. P. *Magariños*, J. F. *Mosquera*, P. *Mutowo*, M. *Nowotka*, et al. "ChEMBL: towards direct deposition of bioassay data". *Nucleic acids research* 47:D1, 2019, pp. D930–D940.

358. P. *He*, X. *Liu*, J. *Gao*, and W. *Chen*. "Deberta: decoding-Enhanced Bert with Disentangled Attention". In: *9th International Conference on Learning Representations, ICLR 2021*. 2021.

359. Z. *Dai*, Z. *Yang*, Y. *Yang*, J. G. *Carbonell*, Q. *Le*, and R. *Salakhutdinov*. "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 2978–2988.

360. H. *Bai*, P. *Shi*, J. *Lin*, Y. *Xie*, L. *Tan*, K. *Xiong*, W. *Gao*, and M. *Li*. "Segatron: Segment-aware transformer for language modeling and understanding". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 12526–12534.

361. Y.-A. *Wang* and Y.-N. *Chen*. "What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding". In: *EMNLP*. 2020, pp. 6840–6849.

362. J. *Bajorath*. "Computational scaffold hopping: cornerstone for the future of drug design?" *Future Medicinal Chemistry* 9:7, 2017, pp. 629–631.

363. J. *Vig*, A. *Madani*, L. R. *Varshney*, C. *Xiong*, R. *Socher*, and N. F. *Rajani*. "BERTology Meets Biology: Interpreting Attention in Protein Language Models". In: *9th International Conference on Learning Representations, ICLR 2021*. 2021.

364. J. *Vig*. "A Multiscale Visualization of Attention in the Transformer Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2019, pp. 37–42.

365. B. *Fabian*, T. *Edlich*, H. *Gaspar*, M. *Segler*, J. *Meyers*, M. *Fiscato*, and M. *Ahmed*. "Molecular representation learning with language models and domain-relevant auxiliary tasks". *arXiv preprint arXiv:2011.13230*, 2020.

366. M. J. *Kusner*, B. *Paige*, and J. M. *Hernández-Lobato*. "Grammar variational autoencoder". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1945–1954.

367. P. *Ertl* and A. *Schuffenhauer*. "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions". *Journal of cheminformatics* 1:1, 2009, p. 8.

368. H. *Boman*. "Antibacterial peptides: basic facts and emerging concepts". *Journal of internal medicine* 254:3, 2003, pp. 197–215.

369. T. U. *Consortium*. "UniProt: the universal protein knowledgebase in 2021". *Nucleic Acids Research* 49:D1, 2020, pp. D480–D489. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa1100. URL: https://doi.org/10.1093/nar/gkaa1100.

370. K. S. *Sarkisyan*, D. A. *Bolotin*, M. V. *Meer*, D. R. *Usmanova*, A. S. *Mishin*, G. V. *Sharonov*, D. N. *Ivankov*, N. G. *Bozhanova*, M. S. *Baranov*, O. *Soylemez*, et al. "Local fitness landscape of the green fluorescent protein". *Nature* 533:7603, 2016, p. 397.

371. G. J. *Rocklin*, T. M. *Chidyausiku*, I. *Goreshnik*, A. *Ford*, S. *Houliston*, A. *Lemak*, L. *Carter*, R. *Ravichandran*, V. K. *Mulligan*, A. *Chevalier*, et al. "Global analysis of protein folding using massively parallel design, synthesis, and testing". *Science* 357:6347, 2017, pp. 168–175.

372. E. C. *Alley*, G. *Khimulya*, S. *Biswas*, M. *AlQuraishi*, and G. M. *Church*. "Unified rational protein engineering with sequence-based deep representation learning". *Nature methods* 16:12, 2019, pp. 1315–1322.

373. S. *Unsal*, H. *Atas*, M. *Albayrak*, K. *Turhan*, A. C. *Acar*, and T. *Doğan*. "Learning functional properties of proteins with language models". *Nature Machine Intelligence* 4:3, 2022, pp. 227–245.

374. D. *Lowe*. "Chemical reactions from US patents (1976 - Sep2016).", 2017. URL: https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873.

375. D. T. *Ahneman*, J. G. *Estrada*, S. *Lin*, S. D. *Dreher*, and A. G. *Doyle*. "Predicting reaction performance in C–N cross-coupling using machine learning". *Science* 360:6385, 2018, pp. 186–190.

376. D. *Perera*, J. W. *Tucker*, S. *Brahmbhatt*, C. J. *Helal*, A. *Chong*, W. *Farrell*, P. *Richardson*, and N. W. *Sach*. "A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow". *Science* 359:6374, 2018, pp. 429–434.

377. F. *Sandfort*, F. *Strieth-Kalthoff*, M. *Kühnemund*, C. *Beecks*, and F. *Glorius*. "A structure-based platform for predicting chemical reactivity". *Chem* 6:6, 2020, pp. 1379–1390.

378. V. *Sanh*, A. *Webson*, C. *Raffel*, S. *Bach*, L. *Sutawika*, Z. *Alyafeai*, A. *Chaffin*, A. *Stiegler*, T. *Scao*, A. *Raja*, et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *International Conference on Learning Representations*. 2022.

379. T. *Brown*, B. *Mann*, N. *Ryder*, M. *Subbiah*, J. D. *Kaplan*, P. *Dhariwal*, A. *Neelakantan*, P. *Shyam*, G. *Sastry*, A. *Askell*, S. *Agarwal*, A. *Herbert-Voss*, G. *Krueger*, T. *Henighan*, R. *Child*, A. *Ramesh*, D. *Ziegler*, J. *Wu*, C. *Winter*, C. *Hesse*, M. *Chen*, E. *Sigler*, M. *Litwin*, S. *Gray*, B. *Chess*, J. *Clark*, C. *Berner*, S. *McCandlish*, A. *Radford*, I. *Sutskever*, and D. *Amodei*. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901.

380. C. W. *Coley*, N. S. *Eyke*, and K. F. *Jensen*. "Autonomous discovery in the chemical sciences part II: outlook". *Angewandte Chemie International Edition* 59:52, 2020, pp. 23414–23436.

381. M. *Benhenda*. "Can AI reproduce observed chemical diversity?" *bioRxiv*, 2018, p. 292177.

382. K. *Preuer*, P. *Renz*, T. *Unterthiner*, S. *Hochreiter*, and G. *Klambauer*. "Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery". *Journal of chemical information and modeling* 58:9, 2018, pp. 1736–1741.

383. P. *Renz*, D. *Van Rompaey*, J. K. *Wegner*, S. *Hochreiter*, and G. *Klambauer*. "On failure modes in molecule generation and optimization". *Drug Discovery Today: Technologies* 32, 2019, pp. 55–63.

384. T. *Cieplinski*, T. *Danel*, S. *Podlewska*, and S. *Jastrzebski*. "We should at least be able to design molecules that dock well". *arXiv preprint arXiv:2006.16955*, 2020.

385. G. A. *Tadesse*, J. *Born*, C. *Cintas*, M. *Manica*, and K. *Weldemariam*. "MPEGO: A toolkit for multi-level performance evaluation of generative models for material discovery domains". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2022.

386. P. S. *Gromski*, A. B. *Henson*, J. M. *Granda*, and L. *Cronin*. "How to explore chemical space using algorithms and automation". *Nature Reviews Chemistry* 3:2, 2019, pp. 119–128.

387. C. W. *Coley*, W. H. *Green*, and K. F. *Jensen*. "Machine learning in computer-aided synthesis planning". *Accounts of chemical research* 51:5, 2018, pp. 1281–1289.

388. M. H. *Segler*, M. *Preuss*, and M. P. *Waller*. "Planning chemical syntheses with deep neural networks and symbolic AI". *Nature* 555:7698, 2018, p. 604.

389. B. *Burger*, P. M. *Maffettone*, V. V. *Gusev*, C. M. *Aitchison*, Y. *Bai*, X. *Wang*, X. *Li*, B. M. *Alston*, B. *Li*, R. *Clowes*, et al. "A mobile robotic chemist". *Nature* 583:7815, 2020, pp. 237–241.

390. C. W. *Coley*, D. A. *Thomas III*, J. A. *Lummiss*, J. N. *Jaworski*, C. P. *Breen*, V. *Schultz*, T. *Hart*, J. S. *Fishman*, L. *Rogers*, H. *Gao*, et al. "A robotic platform for flow synthesis of organic compounds informed by AI planning". *Science* 365:6453, 2019, eaax1566.

391. D. *Probst*, M. *Manica*, Y. G. *Nana Teukam*, A. *Castrogiovanni*, F. *Paratore*, and T. *Laino*. "Biocatalysed synthesis planning using data-driven learning". *Nature communications* 13:1, 2022, pp. 1–11.

392. H. *Tao*, T. *Wu*, S. *Kheiri*, M. *Aldeghi*, A. *Aspuru-Guzik*, and E. *Kumacheva*. "Self-Driving Platform for Metal Nanoparticle Synthesis: Combining Microfluidics and Machine Learning". *Advanced Functional Materials* 31:51, 2021, p. 2106725.

393. M. *Christensen*, L. P. *Yunker*, F. *Adedeji*, F. *Häse*, L. M. *Roch*, T. *Gensch*, G. dos *Passos Gomes*, T. *Zepel*, M. S. *Sigman*, A. *Aspuru-Guzik*, et al. "Data-science driven autonomous process optimization". *Communications Chemistry* 4:1, 2021, pp. 1–12.

394. S. *Langner*, F. *Häse*, J. D. *Perea*, T. *Stubhan*, J. *Hauch*, L. M. *Roch*, T. *Heumueller*, A. *Aspuru-Guzik*, and C. J. *Brabec*. "Beyond ternary OPV: high-throughput experimentation and self-driving laboratories optimize multicomponent systems". *Advanced Materials* 32:14, 2020, p. 1907801.

395. B. *Lester*, R. *Al-Rfou*, and N. *Constant*. "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*. Association for Computational Linguistics, 2021, pp. 3045–3059.

396. C. *Edwards*, T. *Lai*, K. *Ros*, G. *Honke*, and H. *Ji*. "Translation between Molecules and Natural Language". *arXiv preprint arXiv:2204.11817*, 2022.

397. Z. *Zeng*, Y. *Yao*, Z. *Liu*, and M. *Sun*. "A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals". *Nature communications* 13:1, 2022, pp. 1–11.

398. C. *Raffel*, N. *Shazeer*, A. *Roberts*, K. *Lee*, S. *Narang*, M. *Matena*, Y. *Zhou*, W. *Li*, P. J. *Liu*, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *J. Mach. Learn. Res.* 21:140, 2020, pp. 1–67.

399. A. *Chowdhery*, S. *Narang*, J. *Devlin*, M. *Bosma*, G. *Mishra*, A. *Roberts*, P. *Barham*, H. W. *Chung*, C. *Sutton*, S. *Gehrmann*, et al. "Palm: Scaling language modeling with pathways". *arXiv preprint arXiv:2204.02311*, 2022.

400. D. A. *Thomas*, T. A. *Fleming*, and M.-I. *Bittner*. "Implementing fully automated kinase inhibitor characterization using a robotic system". *Cancer Research* 81:13_Supplement, 2021, pp. 295–295.