# TARGETED DRUG DESIGN WITH WARM START

by

Gökçe Uludoğan

B.S., Computer Engineering, Boğaziçi University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2021

# ACKNOWLEDGEMENTS

First of all, I would like to thank my advisors: Assoc. Prof. Arzucan Özgür and Assoc. Prof. Elif Özkırımlı for their guidance, support and patience during my studies.

I would like to express my gratitude to my jury members, Assist. Prof. Fatma Başak Aydemir, Prof. Olcay Taner Yıldız for kindly accepting to participate in the defense of this thesis.

I would also like to thank the members of TABILAB for discussions, small talks and the lovely time spent together in the lab. I would like to extend my sincere thanks to our next-door neighbor, Dr. Suzan Üsküdarlı.

I would like to offer my special thanks to Prof. Dr. Nilgün Karalı, Prof. Dr. Kutlu Ülgen and the members of KimDil project group for their comments and suggestions.

I am grateful to my parents for always supporting me and giving me the opportunities that have made me who I am.

I gratefully acknowledge TÜBİTAK BİDEB 2210/A scholarship program. I also thank TÜBİTAK ARDEB 1001 Program (Project No: 119E133) and TAM project for providing the computational resources.

# ABSTRACT

## TARGETED DRUG DESIGN WITH WARM START

The generation of novel compounds targeting a protein of interest is a compelling task in the pharmaceutical industry. Deep generative models have been applied to targeted molecular design and have shown promising results. However, such models are often limited by the availability of the data they rely on such as protein structure or protein-ligand binding affinity. Notwithstanding, vast amounts of unlabeled protein sequences and chemical compounds are available and have been used to train models which learn useful representations. To transfer this knowledge to targeted drug design, we propose using warm start strategy to initialize models with those pretrained models. We investigate two warm start strategies: (i) one-stage strategy where the initialized model is trained on targeted molecule generation (ii) two stage strategy containing a pre-finetuning on molecular generation followed by target specific training. We also use two decoding strategies to generate compounds: beam search and sampling. The results show that the warm-started models perform better than a baseline model trained from scratch on different percentages of data and decoding strategies. The proposed warm starting strategies obtain similar results in terms of widely used metrics from benchmarks. However, docking evaluation of the generated compounds for a set of novel proteins suggests that the one stage strategy generalizes better than the two stage strategy. Additionally, we observe that beam search outperforms sampling in both docking evaluation and benchmark metrics assessing the quality of compounds.

# ÖZET

## SICAK BAŞLANGIÇ İLE HEDEF ODAKLI İLAÇ TASARIMI

İlgilenilen bir proteini hedefleyen yeni moleküllerin üretilmesi, farmasötik endüstrisindeki zorlayıcı görevlerdendir. Derin üretici modeller, hedef odaklı molekül tasarımı problemine uygulanmış ve umut verici sonuçlar elde edilmiştir. Fakat, genellikle dayandıkları protein yapısı veya protein-ligand bağlılık ilgisi verilerinin miktarı bu tür modellerin başarısını sınırlamaktadır. Bununla birlikte, büyük miktarlarda etiketlenmemiş protein dizileri ve moleküller mevcuttur ve bu verileri kullanarak faydalı temsiller öğrenen modeller eğitilmiştir. Bu tezde, bu bilgiyi hedef odaklı ilaç tasarımına aktarmak için, önceden eğitilmiş modellerin ağırlıklarını, sıcak başlangıç (warm-start) stratejisi ile hedef odaklı modelleri başlatmak için kullanmayı önerdik. İki sıcak başlangıç stratejisini araştırdık: (i) başlatılan modelin hedeflenen molekül üretimi üzerinde eğitildiği bir aşamalı strateji (ii) moleküler üretim üzerinde bir ön ince ayar ve ardından hedefe özel eğitim içeren iki aşamalı strateji. Molekülleri oluşturmak için kullandığımz iki kod çözme stratejisi ışın araması (beam search) ve örneklemedir (sampling). Sonuçlar, sıcak başlangıçlı modellerin, farklı veri miktarları ve kod çözme stratejilerinde sıfırdan eğitilmiş bir modelden daha iyi performans sergilediğini göstermektedir. Sıcak başlangıç stratejileri, yaygın kullanılan karşılaştırma metrikleri açısından benzer sonuçlar elde etmektedir; bununla birlikte, bir dizi yeni protein için üretilen moleküllerin kenetlenme değerlendirmesi, bir aşamalı stratejinin iki aşamalı stratejiden daha genellenebilir olduğunu önermektedir. Ek olarak, ışın aramasının hem kenetlenme değerlendirmesinde hem de moleküllerin kalitesini değerlendiren kıyaslama ölçütlerinde örneklemeden daha iyi performans gösterdiği gözlemlenmiştir.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xii
LIST OF SYMBOLS . . . xiii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiv
1. INTRODUCTION . . . 1
2. BACKGROUND . . . 5
2.1. Drug Discovery . . . 5
2.2. Ligands . . . 5
2.3. Proteins . . . 7
2.3.1. Protein Sequence Similarity . . . 7
2.4. Databases . . . 7
2.4.1. BindingDB . . . 7
2.4.2. Pfam . . . 8
2.4.3. Protein Data Bank (PDB) . . . 8
2.5. Molecular Docking . . . 8
2.6. Technical Background . . . 9
2.6.1. Identification of Biochemical Words . . . 9
2.6.2. Sequence Generation . . . 9
2.6.3. Deep Generative Models . . . 10
2.6.4. Encoder Decoder Models . . . 10
2.6.5. Transformer . . . 12
2.6.6. Self Supervised Pretraining in NLP . . . 12
2.7. Related Work . . . 13
3. MATERIALS AND METHODS . . . 15
3.1. Data . . . 15

3.2. Representation . . . . . 17
3.3. Models . . . . . 17
3.4. Evaluation . . . . . 20
3.4.1. Benchmarking metrics . . . . . 20
3.4.2. Docking . . . . . 21
3.4.3. Synthesizability . . . . . 21
4. RESULTS . . . . . 22
4.1. ChemBERTaLM . . . . . 22
4.2. Target specific models . . . . . 23
4.2.1. Benchmarking metrics . . . . . 24
4.2.2. Docking . . . . . 29
4.3. Synthesizability . . . . . 35
5. DISCUSSION . . . . . 38
6. CONCLUSION . . . . . 40
REFERENCES . . . . . 41
APPENDIX A: CHEMICAL SPACE . . . . . 53
APPENDIX B: DOCKING RESULTS . . . . . 55
APPENDIX C: RETROSYNTHESIS PLANNING . . . . . 60

# LIST OF FIGURES

Figure 2.1. Drug discovery and development stages and the approximate amount of time each stage takes. . . . . 5

Figure 3.1. Distributions of protein sequence similarities between and within datasets. . . . . 16

Figure 4.1. t-SNE plot of generated compounds with EncDecBase model trained on all interactions and 5000 randomly sampled compounds from the test set. . . . . 26

Figure 4.2. FCD score distribution of chemical compounds generated for test proteins across models, data levels and decoding strategies along with the distribution of the scores within interacting compounds of train proteins. . . . . 27

Figure 4.3. SNN distribution of chemical compounds generated for test proteins across models, data levels and decoding strategies along with the distribution of the scores within interacting compounds of train proteins. . . . . 28

Figure 4.4. ROC Curves for the comparison of active compounds to random ones for each protein structure. AUC scores and p values (Mann-Whitney's U test) are reported in parenthesis respectively. . . . . 30

Figure 4.5. ROC Curves for the comparison of the generated compounds for MAP3K14 with random compounds and active compounds. The labels indicate the model type, the data level the model trained on and the decoding strategy. . . . . 32

Figure 4.6. ROC Curves for the comparison of the generated compounds for PLK3 with random compounds and active compounds.The labels indicate the model type, the data level the model trained on and the decoding strategy. . . . . 33

Figure 4.7. Synthesizability scores of generated compounds by the best performing model across targets. . . . . 36

Figure 4.8. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 4B6L. . . . . 37

Figure A.1. t-SNE plot of generated compounds with EncDecLM model trained on all interactions and 5000 randomly sampled compounds from the test set. . . . . 53

Figure A.2. t-SNE plot of generated compounds with T5 model trained on all interactions and 5000 randomly sampled compounds from the test set. . . . . 54

Figure B.1. ROC Curves for the comparison of the generated compounds for 5XY1 with random compounds and active compounds. . . . . 55

Figure B.2. ROC Curves for the comparison of the generated compounds for 2WO6 with random compounds and active compounds. . . . . 55

Figure B.3. ROC Curves for the comparison of the generated compounds for 1ERE with random compounds and active compounds. . . . . 56

Figure B.4. ROC Curves for the comparison of the generated compounds for 4I23 with random compounds and active compounds. . . . . 56

Figure B.5. ROC Curves for the comparison of the generated compounds for 5K0K with random compounds and active compounds. . . . . 57

Figure B.6. ROC Curves for the comparison of the generated compounds for 5TUY with random compounds and active compounds. . . . . 57

Figure B.7. ROC Curves for the comparison of the generated compounds for 5V1B with random compounds and active compounds. . . . . 58

Figure B.8. ROC Curves for the comparison of the generated compounds for 6LVL with random compounds and active compounds. . . . . 58

Figure B.9. ROC Curves for the comparison of the generated compounds for 6WJ5 with random compounds and active compounds. . . . . 59

Figure C.1. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 2WO6. . . . . 60

Figure C.2. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 4I23. . . . . 60

Figure C.3. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 5K0K. . . . . 61

Figure C.4. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 6LVL. . . . . 61

Figure C.5. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 6WJ5. . . . . 62

Figure C.6. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 6Z1Q. . . . . 62

# LIST OF TABLES

Table 3.1. Statistics of the data set extracted from BindingDB. . . . . 15

Table 3.2. Summary of data splits. . . . . 16

Table 4.1. Performance metrics for ChemBERTaLM and baseline models: fraction of valid compounds, fraction of compounds passing filters, internal diversity and novelty. . . . . 23

Table 4.2. Performance metrics for ChemBERTaLM and baseline models on random test set (Test) and scaffold split test set (TestSF) from MOSES benchmark. . . . . 23

Table 4.3. Performance metrics for target specific models on different percentages of data and decoding strategies. . . . . 24

Table 4.4. Jensen Shannon distance between FCD scores of the compounds generated for the test proteins and the scores of the compounds interacting with the train proteins. . . . . 28

Table 4.5. Number of proteins in which generated compounds can be distinguished from random compounds for each model and decoding strategy. . . . . 34

Table 4.6. Comparison of activity against each target between the active compounds, the generated ones for the target and the generated ones for the others. . . . . 35

# LIST OF SYMBOLS

| | |
|---|---|
| c | Hidden states |
| $EC_{50}$ | Half maximal effective concentration |
| $IC_{50}$ | Half maximal inhibitory concentration |
| $K_i$ | Inhibition Constant |
| $K_d$ | Dissociation Constant |
| $x$ | Input sequence |
| $y$ | Output sequence |
| $y^*$ | Most probable output sequence |
| $K$ | Key matrix |
| $Q$ | Query matrix |
| $V$ | Value matrix |
| | |
| $\theta_{enc}$ | Parameters of encoder |
| $\theta_{dec}$ | Parameters of decoder |

# LIST OF ACRONYMS/ABBREVIATIONS

| | |
|---|---|
| 1D | One Dimensional |
| 2D | Two Dimensional |
| 3D | Three Dimensional |
| AUROC | Area Under the Receiver Operating Characteristics |
| BO | Bayesian Optimization |
| BPE | Byte Pair Encoding |
| D | Dimension |
| Frag | Fragment Similarity |
| FCD | Fréchet ChemNet Distance |
| GAN | Generative adversarial network |
| ID | Identifier |
| IntDiv | Internal Diversity |
| K | Thousand |
| M | Million |
| NLP | Natural Language Processing |
| nM | Nanomolar |
| PDB | Protein Data Bank |
| RL | Reinforcement Learning |
| RNN | Recurrent neural network |
| Scaf | Scaffold Similariy |
| SMILES | Simplified Molecular Input Line Entry System |
| SNN | Similarity to a nearest neighbor |
| TL | Transfer Learning |
| t-SNE | t-Distributed Stochastic Neighbor Embedding |
| UniProt | The Universal Protein Resource |
| VAE | Variational Autoencoder |

# 1. INTRODUCTION

The discovery of novel chemical entities with desired attributes is a compelling task in drug discovery. To find chemical compounds biased towards a biological target, candidate compounds must be identified from the vast space of the potential drug-like molecules, which has been estimated to be on the order of $10^{60}$ [1]. Although the advances in the high throughput screening allow screening of a large number of compounds against a biological target, these experiments are expensive and the space is still too large for screening. Another challenge arises from the promiscuity of small drug molecules, which is the ability to interact with many targets. It is shown that drugs designed for a specific target are active towards 11 other targets on average [2]. This could lead to unexpected toxicity, one of the major causes of attrition in drug discovery [3]. These issues along with the complexity of human biology offer a challenge and opportunity for developing scalable and efficient tools for drug discovery.

To speed up drug discovery pipeline and minimize costs, computational methods have been utilized extensively. Virtual screening aims to identify active compounds towards a biological target by screening large chemical libraries. Thus, this approach is restricted to the synthetically accessible molecules, which constitute only a small portion of theoretically possible drug-like space comprising more than $10^{60}$ molecules. De novo drug design stands out as an efficient alternative to screening methods. The goal in de novo drug design is to generate new chemical entities with particular properties of interest from scratch. This approach enables the exploration of untapped chemical space efficiently while avoiding brute-force screening.

De novo design approaches can be broadly divided into structure-based drug design and ligand-based drug design [4]. When the three-dimensional structure (3D) of a target of interest is available, structure-based design approaches can be applied. On the other hand, ligand-based drug design methods exploit the knowledge of active and inactive molecules towards the targets. Traditional methods for de novo drug design

are based on mostly simulations and heuristics, and also require significant expert knowledge. To overcome the limitations, deep generative models have been applied to de novo drug design [5–13]. These models are data-driven and allow navigation of the chemical space by sampling. However, existing models have certain drawbacks. Structure based methods are limited by the availability of 3D structure of proteins. As of June 2021, the available protein structures are much smaller than the number of protein sequences (Protein Data Bank, [14] contains 157K structures while UniProt [15] comprises 219M sequences). On the other hand, the protein-ligand interactions required by ligand-based methods are relatively small compared to unlabeled sequences. Notwithstanding, the vast amounts of such sequences have been used to train models which learn useful representations and this knowledge learned by these models has transferred to other tasks.

In this study, we propose warm start strategy to initialize targeted models with pretrained models on a large scale of sequences. Given that the interacting protein-compound pairs with reported affinity are limited, we suggest that using pre-trained models to initialize the targeted model might allow the model to benefit from the representations learned on diverse sequences, and thus, enhance generalizability and boost performance. In addition, warm start strategy could save significant computational time.

We view the target specific molecule generation as a translation task from protein language to chemical language. Similar to human language, proteins and chemical compounds can be represented as sequences. Moreover, proteins are composed of functional units shared between different proteins. Likewise, chemical compounds consist of subfragments which have been shown to follow power law similarly to natural languages [16]. Considering these analogies between biochemical languages and human languages, many studies adopted methods from natural language processing (NLP) to process biochemical sequences and identify meaningful units of these languages [17–21]. Recent studies have applied subword segmentation algorithms to split the sequences into tokens and shown that this approach achieves superior performance compared to

character level tokens [22,23]. Therefore, we consider protein and chemical units identified by subword segmentation algorithms as tokens rather than individual characters in sequences. This reduces the length of sequences, and therefore, allows better capturing of long-range dependencies which is crucial for this problem.

We employ a Transformer based sequence-to-sequence model and initialize the encoder and the decoder components with pretrained models on large-scale data. For the encoder, we adopt a protein RoBERTa model [19] trained on a mixture of binding and nonbinding protein pairs from STRING database [24]. We choose this model for two reasons: First, it is well suited to our formulation as it uses Byte-Pair Encoding (BPE) algorithm [25] to split protein sequences into "protein words". Second, by training with protein pairs, the model has achieved promising scores in binding tasks and the captured features might be beneficial for target specific molecule generation. For the initialization of the decoder, we adopt ChemBERTa model [18] trained on 10M SMILES from PubChem [26]. Similar to the protein model, this model employs BPE algorithm to identify "chemical words".

We test two warm start strategies: first, fine-tuning of the initialized model with protein-ligand pairs filtered from BindingDB [27], and second, an initial fine-tuning with compounds from MOSES [28] followed by training with interacting protein-compounds pairs. To measure the effect of warm starting, we create subsets of data containing 5%, 25%, and 100% of the interactions in our data set, respectively. We suggest if warm-started models perform well in low data regimes, the strategy can be used to train targeted models for particular protein families where proteins within a family are involved in diverse processes and selective compounds for such proteins are desirable. To compare with the warm-started models, we adopt T5 model [29], a Transformer model slightly different from the model used in previous work [30]. We train these models for each combination of strategies and levels of data. We also compare two decoding strategies for the generation of molecules: beam search and sampling.

We evaluate the generated compounds with metrics from the MOSES benchmark and by performing docking. The results show that the warm-started models perform better than the T5 model trained from scratch on different percentages of data and decoding strategies. The two proposed warm starting strategies obtain similar scores to each other with respect to metrics from the MOSES benchmark; however, docking evaluation of the designed compounds for a set of novel proteins suggests that the one stage strategy generalizes better than the two stage strategy. Additionally, we observe that beam search outperforms sampling in both docking evaluation and benchmark metrics assessing the quality of the generated chemical molecules.

# 2. BACKGROUND

## 2.1. Drug Discovery

Drug discovery and development is an expensive, time-consuming, and risky process comprising many stages [31]. The procedure starts with the identification of a target for a drug to act on. Then, this target is validated by demonstrating that it is involved in the disease and the modulation of the target is likely to have a therapeutic effect. This is followed by searching for chemical compounds showing activity against this target. Promising compounds called *leads* are chosen and optimized for certain physicochemical attributes and potency against the target. Next, the candidates are investigated in vitro (outside of living organisms) and in vivo (inside living organisms) experiments. The successful candidates are assessed in clinical trials. If the drug is approved, then it is brought into the market. These stages and the amount of time these stages take are summarized in Figure 2.1 [31].

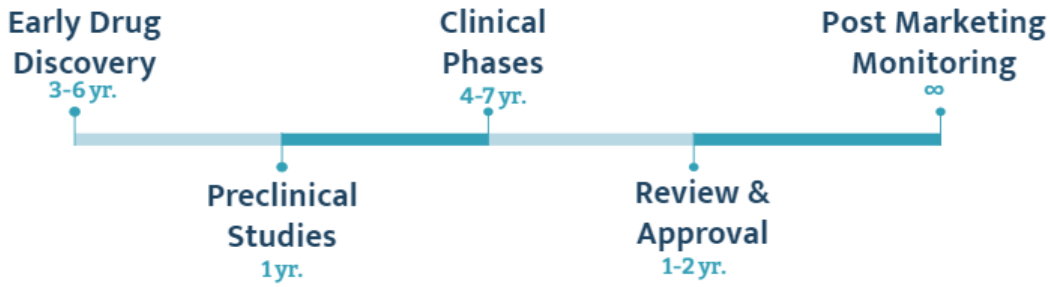


Figure 2.1. Drug discovery and development stages and the approximate amount of time each stage takes.

## 2.2. Ligands

A ligand is defined as any substance which forms a complex with biomolecules. However, in this thesis, we are interested in small molecules as ligands since those molecules are more likely to be chemically stable and satisfy physicochemical characteristics. Such molecules form 90% of drugs [32].

There are many ways to represent a molecule with different forms (1D, 2D, 3D). The simplest and well-known one-dimensional representation is the molecular formula where the molecule is described by the elements and the counts of the elements. Another one-dimensional representation, which is especially widely used in machine learning applications, is the Simplified molecular-input line-entry system (SMILES). SMILES notation is a way of representing a 2D molecular graph in one dimension and this is obtained by a traversal of 2D molecular graph. Atom types, bond types, branches, and rings are denoted with specific symbols in SMILES strings. In this notation, one molecule can be represented with multiple SMILES strings. On the one hand, this could be used for data augmentation. On the other hand, it might also lead to ambiguity. In cases where unique SMILES is needed to represent a molecule, this can be obtained with the canonicalization process and the yielded SMILES is called canonical SMILES.

Aside from 1D representations, molecules can also be represented with 2D molecular graphs. In this regard, a molecule is viewed as a graph where each atom is a node and each bond is an edge. This representation has received much attention recently [33–36]. However, processing graph representation is computationally expensive compared to textual representations. Considering the structures of molecules, in reality, 3D representation is the intrinsic way of representing molecules. In this representation, atoms are represented with their positions in 3D space.

Fingerprints are among the classical representations for molecules. There are different categories of fingerprints relying on pre-defined rules or hashing. Extended-Connectivity Fingerprint (ECFP) [37] is a well-known hash-based fingerprint. This method generates the substructure patterns by considering the circular neighborhood of each atom up to radius $n$ and encode the presence of these structures.

## 2.3. Proteins

Proteins are crucial elements of life and our biology. They have essential roles in the cell, including structural, biochemical, and signaling functions [38]. Proteins can be represented with different levels of structures. The primary structure of a protein is the sequence of amino acids in its chains. These amino acid residues are connected with peptide bonds. Higher levels of protein structures - secondary, tertiary, and quaternary - are 3D forms of proteins. The function of a protein depends on its 3D structure [38]. However, the 3D structure of proteins is limited compared to the primary structures. Given that the 3D structure of a protein is determined by its amino acid sequence and a large amount of protein sequences available, representing proteins with their sequences is suitable.

There are 20 amino acid residues that make up proteins. Each amino acid residue has a unique three-letter code and one-letter code. The protein sequences range in length from tens to several thousand amino acids.

### 2.3.1. Protein Sequence Similarity

Needleman Wunsch (NW) [39] is an algorithm for finding optimal global alignment between two protein sequences. The goal is to determine the sequence similarity between these proteins.

## 2.4. Databases

### 2.4.1. BindingDB

BindingDB [27] is a public database of protein-ligand interactions. This database provides measured binding affinities of protein-ligand pairs. Each data entry in this database includes the sequence of the target, SMILES representation of the ligand, the measured affinity, and the source of this experiment.

### 2.4.2. Pfam

Pfam [40] is a database containing protein families, domains and their multiple sequence alignments built using Hidden Markov Model (HMM). Protein families are sets of proteins that share an evolutionary origin. Proteins within a family typically have similar structures/sequences or functions.

### 2.4.3. Protein Data Bank (PDB)

Protein Data Bank (PDB) [41] stores three-dimensional structures of large biological molecules. As of June 2021, it contains around 179K entries. Each PDB entry is assigned a four-letter accession code and consists of the coordinates of proteins with solvent and bound molecules. The structures of proteins or protein-ligand complexes can be downloaded from PDB.

## 2.5. Molecular Docking

Molecular docking is a computational method that predicts the conformation and the binding affinity of a molecule binding to a protein. Docking comprises of two stages: sampling (search) and scoring. The sampling stage involves the exhaustive search of the conformational space for the protein-molecule pair. Then, sampled poses are assessed with the scoring function. The classical scoring functions are divided into three classes: physics-based, knowledge-based, and empirical [42]. With the substantial increase in the number of available protein structures, machine learning (ML) algorithms are used as scoring functions. ML based functions have achieved comparable or better performance compared to the classical ones [42]. GNINA [42] is deep learning based molecular docking tool forked from SMINA [43] and AutoDock Vina [44], docking tools with empirical scoring function. By default, GNINA only uses deep learning based scoring for the last ranking of the ligand conformations. However, such scoring can be utilized in other steps as well with additional computational cost.

## 2.6. Technical Background

### 2.6.1. Identification of Biochemical Words

Biochemical entities such as proteins and chemical compounds can be represented as strings of characters. We use primary structure (i.e. amino acid sequence) to represent proteins and SMILES strings to describe molecules. These representations of the entities can be considered as domain specific languages. In this sense, a protein sequence or SMILES string is analogous to a sentence. We assume that these sentences are composed of meaningful units similar to words in human language. To extract such so-called words, we adopt the Byte-Pair Encoding algorithm.

Byte-Pair Encoding is a compression technique [45] which has been adopted to NLP for segmenting sentences [25]. The algorithm starts with an initial vocabulary consisting of characters from the corpus. Then, character pairs in the corpus are counted and the most frequent ones are merged and added to the vocabulary. This iteration continues until desired vocabulary size is obtained or a particular number of the merge operation is completed.

BPE algorithm has been employed by recent studies to identify protein and chemical words for various tasks [22,46,47]. This method has achieved competitive or better performance compared to other tokenizations [22]. In addition, it also reduces computational costs by shortening sequences. This is crucial for especially proteins which are made up of long amino acid sequences.

### 2.6.2. Sequence Generation

The task of sequence generation can be formulated as learning distribution $p(x)$ of sequence $x$ given a sequence of $x = (x_1, x_2, \ldots, x_n)$ where each $x_i$ belong to the vocabulary V. Similarly, conditional sequence generation can be defined as modeling distribution $p(y|x)$ given an input sequence $x$ and an output sequence $y$.

### 2.6.3. Deep Generative Models

Deep generative models learn the underlying distribution of given data and generate new samples from the same distribution. These models can be classified into explicit density and implicit density models depending on whether or not the distribution is explicitly specified.

Autoregressive and latent variable models are notable explicit density models. In autoregressive models, the probability distribution is modeled by factorization into a product of conditional probabilities using the chain rule. The models are optimized with maximizing the likelihood of training data. On contrary, latent variable models define a function by introducing a latent variable $z$. Since directly maximizing this likelihood is difficult, the models are trained on the derived lower bound of the likelihood.

For the conditional generation, autoregressive models decompose the distribution $p(y|x)$ into a product of conditional distributions where $x$ denotes the input sequence and $y$ denotes the output sequence. The likelihood is defined as:

$$p(y \mid x) = \prod_{i=1}^{T} p\left(y_i \mid y_{<i}, x\right). \tag{2.1}$$

In this formulation, the likelihood of an output sequence conditioned on an input sentence is equal to the product of the probability of each item in the output sequence given the input sequence and previous items of the output sequence.

### 2.6.4. Encoder Decoder Models

Encoder decoder models, also known as sequence-to-sequence models, are conditional generative models. As the name suggests, it is used for sequence-to-sequence models such as machine translation. The model is composed of two components: an encoder and a decoder. The encoder transforms input to its hidden state representation.

Then, the decoder uses the encoder hidden states to generate an output sequence. Previously these components were built by stacking Recurrent Neural Networks (RNN). However, because of the recurrent nature, such model is not parallelizable and cannot capture long range dependencies. With the introduction of Transformer model [48] addressing such issues, Transformer blocks have become the standard components of the encoder decoder models.

Formally, the encoder learns to map input $x$ to hidden states $c$: $f_{\theta_{\text{enc}}} : \mathbf{x} \rightarrow \mathbf{c}$. The probability of an output sequence $y$ is defined by the decoder as:

$$p_{\theta_{\text{dec}}}\left(\mathbf{y}_{1:T} \mid \mathbf{c}\right) = \prod_{i=1}^{T} p_{\theta_{\text{dec}}}\left(\mathbf{y}_i \mid \mathbf{y}_{0:i-1}, \mathbf{c}\right). \tag{2.2}$$

Encoder decoder models can be trained autoregressively to predict the next token of the output sequence given the input sequence and previous output tokens. Given an input sequence $x$, the most probable output sequence $y^*$ is formally defined as:

$$y^* = \arg\max_{y} p(y \mid x). \tag{2.3}$$

Finding the exact solution is not feasible since the space of possible output sequences is so large. Thus, decoding strategies are used to generate output sequences.

The straightforward way of generating output sequence is selecting the most probable token at each step during decoding. This strategy is known as greedy decoding and might lead to repetitive outputs [49]. Beam search strategy might alleviate this problem by keeping N most probable hypotheses at each step. In addition, output sequences can be generated with sampling strategy in which next tokens are chosen randomly according to the conditional distribution.

### 2.6.5. Transformer

Transformer [48] is a model based on encoder-decoder architecture. Encoder and decoder components are composed of stacks of identical blocks. Each encoder block is formed by a self-attention layer followed by a feed-forward neural network. Self attention layer allows the model to look at the input sequence while encoding a specific token. In addition to these layers, each decoder block contains another attention layer (i.e. cross attention/encoder-decoder attention) between them. Like self attention, cross attention enables attending to the input sequence. Note that self attention layers in the decoder are masked to only attend previous outputs of the decoder.

Each attention layer consists of multiple heads and the computations in these heads run in parallel. Each head takes the keys $K$, the values $V$, and the queries $Q$ created for each token and packed into matrices. The attention is computed as follows:

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.4}$$

where $d_k$ is a factor used for scaling depending on the layer size.

Since Transformer model lacks recurrence and convolution, the order of tokens must be injected explicitly to the model. This is achieved by adding a vector indicating the position of token in the sequence to each input embedding. There are different types of positional encodings. The original Transformer uses sinusoidal or learned positional encodings.

### 2.6.6. Self Supervised Pretraining in NLP

Self supervised learning is a type of unsupervised learning where the data itself are used to train models with a proxy task. Self supervised methods have been used to pretrain models in NLP [50,51]. By fine-tuning in downstream tasks, such models have pushed the-state-of-the-art and saved significant time and computational resources.

Bidirectional Encoder Representations from Transformers (BERT) [50] is a model consisting of Transformer Encoder blocks. The model uses two self-supervised objectives: next sentence prediction and masked language modeling. In masked language modeling objective, a fraction of input is masked, and the model is trained to predict masked tokens. BERT model has been pretrained with these objectives and fine-tuned on various tasks and obtained the state-of-the-art results.

RoBERTa (Robustly Optimized BERT Pretraining Approach) [51] is an improved pretraining strategy for BERT model. Compared to BERT, RoBERTa has been trained with only masked language modeling and optimized hyperparameters.

## 2.7. Related Work

Deep generative models have been applied to molecular design. Existing models can be classified into two categories depending on the usage of the target information.

Earlier models use only the knowledge of the compounds similar to ligand-based design. In these works, the general approach is training a model with a library of compounds to learn the chemical space and sampling from this model to generate new chemical entities similar to those used for training. For this purpose, Recurrent neural networks (RNN), variational autoencoders (VAE) and Generative Adversarial Networks (GAN) are employed. To represent molecules, these studies use either Simplified Molecular Input Line Entry System (SMILES) [52] strings or molecular graphs [8, 9, 53–55].

For biased generation of the chemical compounds, molecule generators can be fine-tuned with one of the following approaches: transfer learning (TL) [5, 9, 53], reinforcement learning (RL) [10, 56–58], and bayesian optimization (BO) [59]. In transfer learning, the model is further trained with a set of compounds that possess the desired properties. Thus, to guide a model for targeted generation, this method requires the knowledge of the compounds which have activity against the target of interest. On the

other hand, reinforcement learning and bayesian optimization methods pair the generative model with a predictive model to score molecules and bias generation. However, it is shown that guiding generation by the scores obtained by a predictive model is problematic [60]. Additionally, these methods are computationally costly.

The second category of generative models leverages the structure of the target [6, 7, 61]. These models are conditioned on the protein pockets and attempt to learn the mapping between the binding site and the chemical compounds. However, the major limitation of these models is the scarcity of structural information for targets. Contrary to these studies, [30] proposed to use the sequence of proteins for a targeted molecular generation. In this study, the targeted generation problem is viewed as a translation task between protein and chemical languages and Transformer architecture has been adopted to target specific generations. However, the number of proteins used in training (around 1100 proteins) limits the generalization of this model.

Lately, self-supervised pretraining has become the dominant paradigm in natural language processing (NLP) [50, 51, 62–64]. Self-supervised methods use unlabeled data for training and exploit large amounts of data. Then, the model can be fine-tuned to another task to save computational time and benefit from the representations learned from an enormous amount of data. Combined with large datasets and big models, this paradigm has obtained state-of-the-art results in many NLP tasks [50, 51, 62–64]. The rapidly growing sequence data of biochemical entities and analogies between human language and biochemical languages have encouraged the application of these methods in bio/cheminformatics [**?**, 18, 19, 65]. By viewing protein sequences as sentences and one or several amino acid residues as words, these techniques have been adopted in protein language modeling and shown to learn useful representations and improve the state-of-the-art in a set of protein engineering tasks [20, 66]. Similarly, by treating the SMILES string of each chemical compound as a sentence and each symbol or subsequent symbols as a word, chemical language models have been trained and shown promising results on downstream tasks [18].

# 3. MATERIALS AND METHODS

## 3.1. Data

To train and assess targeted generative models, we used protein-ligand interactions filtered from BindingDB [27] which contains measured binding affinities between proteins and small molecules. First, we filtered out the interactions missing either protein or SMILES sequence. In 95% of the reported interactions, the proteins were assayed as a single chain, so we dropped multichain proteins for simplicity. We used UniProt identifiers of proteins to obtain families from Pfam [40], thus we also excluded the proteins without UniProt identifier or Pfam family. After omitting the interactions not having any affinity measurement, the remaining interactions were labeled as *active* or *inactive* based on affinity scores. BindingDB includes several affinity metrics ($K_i$, $K_d$, $IC_{50}$, $EC_{50}$) which are not directly comparable. To set an affinity threshold, $K_i$ and $K_d$ values are first converted to $IC_{50}$ by a factor 2 by following [67]. Then, we labeled the interactions with affinity scores below 100 nM as active and those with affinity scores above 10000 nM as inactive [68]. For the protein-ligand pairs with multiple reported affinity scores and assay, we calculated the geometric mean of these values and compare this score with the thresholds to label these interactions [67]. The statistics of the resulting data set are reported in Table 3.1.

Table 3.1. Statistics of the data set extracted from BindingDB.

| **Label** | **# Interactions** | **# Unique Proteins** | **# Unique Ligands** |
|---|---|---|---|
| **Active** | 428067 | 3099 | 331942 |
| **Inactive** | 64696 | 3817 | 123763 |

We relied on sequence similarities and Pfam families of proteins while splitting active interactions. The similarities between protein sequences were computed using Needleman-Wunsch global alignment with BioPython [69] wrapper of EMBOSS [70].

We aimed to create diverse validation and test sets. For this purpose, we sampled 10% of proteins in each Pfam family and computed similarities between these proteins and all remaining proteins. Then, the selected proteins were binned based on maximum similarities to the remaining proteins. From these bins, 200 proteins were chosen in total with weighted random sampling based on the inverse frequency of the bins. The interactions of these proteins constitute the validation set. To form the test set, the steps described above were repeated with the remaining proteins. The distributions of sequence similarities between and within these splits are shown in Figure 3.1. The summary of the splits can be seen in Table 3.2

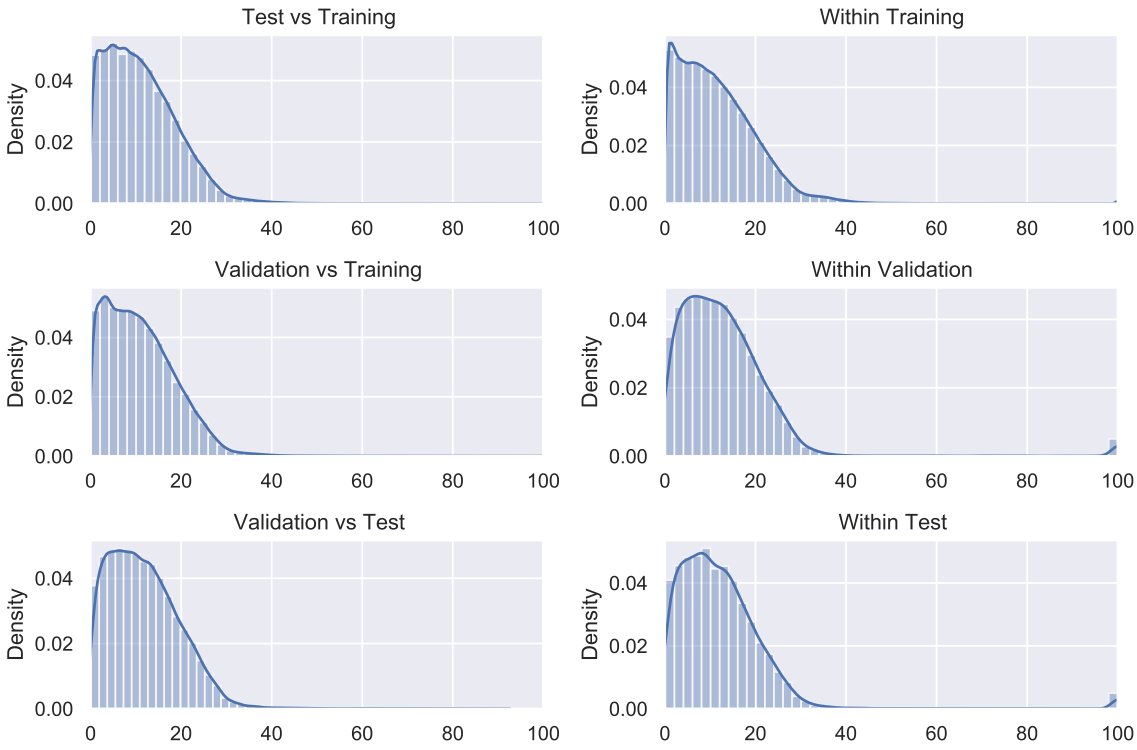


Figure 3.1. Distributions of protein sequence similarities between and within datasets.

Table 3.2. Summary of data splits.

| **Split** | **# Interactions** | **# Unique Proteins** | **# Unique Ligands** |
|---|---|---|---|
| **Training** | 310300 | 2337 | 257348 |
| **Validation** | 25335 | 200 | 24121 |
| **Test** | 21350 | 200 | 20675 |

To compare the effect of warm starting depending on the data size, we also created splits containing only 5% and 25% of the interactions from the splits. To preserve the diversity while reducing the size, we grouped proteins within the splits and sampled interactions based on the inverse of the number of interactions for each protein.

Aside from BindingDB data, we also used MOSES dataset [28] filtered from the ZINC database [71] since the second warm start strategy requires a collection of compounds. This dataset contains 1,936,962 drug-like molecules split into training (1.6M), test (176K), and scaffold test (176K).

### 3.2. Representation

The view of targeted drug design as a translation task relies on the analogy between biochemical languages (i.e. protein language and chemical language) and human languages. In both languages, sentences consist of small reused units. The arrangement of these units determine the semantics/functions of the sentences. In line with this view, proteins and chemical compounds are represented with their so-called words. We used the vocabularies constructed with the BPE algorithm by the pretrained models which we employed for warm starting. The protein vocabulary contains 10K protein words while the chemical vocabulary comprises around 8K chemical words.

### 3.3. Models

We applied a Transformer-based sequence-to-sequence model on targeted de novo design with warm starting from the pretrained models. The model is based on an encoder-decoder architecture which is the prevalent approach for sequence-to-sequence tasks in NLP. In this paradigm, the encoder takes an input sequence and encodes it to a sequence of hidden states while the decoder produces an output sequence autoregressively given the encoder hidden states. We used pretrained Transformer variants for initializing our model to benefit from the representations learned on diverse sequences.

The original Transformer is introduced in [48] and shown to be effective in various NLP tasks and adapted in different domains [18,19,33,50,62,64,72]. The key component of this model is self-attention which allows the model to relate different parts of the sequence while computing the representation of the sequence. Relying on attention blocks makes the model computationally efficient and also helps to capture long-range dependencies. The model is composed of encoder and decoder stacks containing self-attention and feed-forward layers. The self-attention mechanism in the encoder and the decoder is similar, however, the decoder uses a causal attention mask to prevent attending to the next words for an autoregressive generation. In addition, the decoder has cross attention layers that attend to the encoder hidden states.

In this study, we leveraged Transformer variants pretrained on large datasets to initialize models. For the encoder part, we utilized the checkpoints of Protein RoBERTa [19]. Protein RoBERTa model has been pretrained with the masked language modeling using 5M binding/non-binding protein sequence-pairs collected from STRING database [24]. Contrary to other pretrained protein models, this model compresses protein sequences using by Byte-Pair Encoding (BPE) algorithm and supports long sequences without increasing memory requirements. This aligns with our formulation of the problem. Protein RoBERTa uses the encoder part of the Transformers, therefore all parameters required by our model can be transferred from this model.

To initialize the decoder of our model, we adopted ChemBERTa model [18] pretrained on 10M PubChem compounds with BPE tokenization [25]. The model does not achieve the state-of-the-art performance, however; it is shown that it learns better representations as trained with additional data and this information might be leveraged in downstream tasks. Similar to Protein RoBERTa, ChemBERTa also uses the RoBERTa model based on the Transformer encoder and has been pretrained with masked language modeling. As ChemBERTa consists of only encoder blocks, we were able to warm start self-attention and feed-forward layers of the decoder. The cross attention layers are initialized randomly.

For finetuning the encoder-decoder models with warm-starting, we followed two strategies: (1) two-stage fine-tuning where we first finetuned the decoder initialized with ChemBERTa weights on compounds from MOSES dataset to obtain a generic compound generator, namely ChemBERTaLM, and then, we initialized the encoder and decoder with Protein RoBERTa and ChemBERTaLM weights respectively and finetuned the model on the protein-ligand interactions we filtered from BindingDB. We refer to the model using this strategy as *EncDecLM*. (2) one-stage finetuning where we finetuned the encoder-decoder model on the target specific generation by initializing with Protein RoBERTa and ChemBERTa checkpoints. The model is referred to as *EncDecBase*.

To compare the effectiveness of warm starting with training from scratch, we adopted the architecture of the T5 model [29]. This model achieved the state-of-the-art results in generation tasks and differs slightly from the original Transformer which has been used in previous work for protein specific molecule generation [30]. Recent studies showed that training bigger models with early stopping is the best compute-efficient training strategy. For this reason, we build a slightly larger model compared to those in the previous work. Our T5 model uses 4 layers with a hidden size of 256 and a feed-forward layer size of 512 and 6 attention heads.

We implemented all models using HuggingFace's transformers library [73]. ChemBERTaLM was finetuned for 10 epochs with default settings. All other models were fine-tuned using Adam optimizer with a linear learning rate schedule with 2000 warm-up steps followed by a linear decay. The batch size was set to 8 and the gradients were accumulated every 8 steps to obtain an effective batch size of 64. Encoder decoder models (i.e. EncDecBase and EncDecLM) were trained for 10 epochs, except for the model using 100% of data, which we trained for 5 epochs. On the other hand, T5 models were trained from scratch and needed more epochs to converge. These models were trained for 20 epochs on 5% and 25% of data and 10 epochs on 100% of data. Once the models were trained, we used the best epoch of each model (i.e. the one with the best validation loss) to generate chemical compounds.

## 3.4. Evaluation

### 3.4.1. Benchmarking metrics

We used a set of metrics from MOSES benchmark to assess the quality and diversity of the generated chemical compounds.

- **Validity:** The percentage of valid compounds. A compound is considered as valid if it can be parsed by RDKit [74].
- **Uniqueness:** The percentage of unique molecules.
- **Novelty:** The percentage of novel compounds (i.e. not present in the training set).
- **Fréchet ChemNet distance (FCD) [75]:** A distance metric computing chemical and biological similarity between two sets of compounds.
- **Scaffold Similarity:** Cosine similarity of Bemis-Murcko scaffold [76] frequencies between two groups of compounds.
- **Fragment Similarity:** Cosine similarity between frequencies of BRICS fragments [76] of two compound sets.
- **Nearest Neighbor Tanimoto Similarity (SNN):** The average Tanimoto similarity between a set of compounds and their nearest neighbors in another set of compounds.
- **Internal Diversity:** the average Tanimoto distance within a set of compounds where Tanitomo distance is equal to 1 minus Tanimoto similarity. This metric can be computed with powers of Tanimoto similarity. $IntDiv_2$ refers to the average squared Tanimoto distance between compounds.
- **Filters:** The percentage of compounds passing a set of filters.

### 3.4.2. Docking

To assess generated chemical compounds in terms of binding affinity, we performed docking for a set of novel proteins. To measure discriminative ability of the docking tool between sets of compounds, we used Receiver Operating Characteristic (ROC) and the corresponding area under the curve (AUC).

The ROC curve is a probability curve that illustrates the performance of a classifier at different thresholds. It is obtained by plotting True Positive Rate (TPR) against False Positive Rate (FPR). The area under this curve (AUC) serves as a measure of the model's ability to distinguish between classes. True Positive Rate (TPR) and False Positive Rate (FPR) are defined as follows:

$$TPR = \frac{TP}{TP + FN} \tag{3.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.2}$$

where TP, TN, FP, and FN refer to the number of True Positive, True Negative, False Positive and False Negative predictions, respectively.

### 3.4.3. Synthesizability

We also evaluated generated compounds for a set of targets in terms of synthetic accessibility via Molecule.one API.

# 4. RESULTS

In this study, we proposed and investigated warm starting strategy for target specific molecule generation. We tested two warm start strategies to initialize the models. We compared these strategies with each other as well as training from scratch on different subsets of data. Besides, we assessed two decoding methods: beam search and sampling.

## 4.1. ChemBERTaLM

ChemBERTaLM model was trained for warm-starting targeted generative model. Thus, we aimed not to achieve state-of-the-art results but to obtain a comparable model to be able to generate compounds. To assess ChemBERTaLM, we generated 30K compounds by sampling and benchmarked the model using MOSES. Table 4.1 and Table 4.2 report the performances of ChemBERTa model along with baseline models in this benchmark. ChemBERTaLM outperforms the baseline models in terms of validity and performs on par in terms of the fraction of compounds passing filters and internal diversity. Although the model obtain a lower novelty score compared to LatentGAN and JTN-VAE models, it shows the best results in terms of FCD score on the scaffold split test set. This test set is compiled to assess if a model is able to generate chemical compounds with novel scaffolds not present in the training and measures generalizability of the model. The performance of ChemBERTaLM is also comparable to the baselines with respect to other similarity metrics (i.e. SNN, Frag, Scaf) on the test sets. These results indicate ChemBERTaLM generalizes well. We should also note that the ChemBERTaLM model is fine-tuned for only 10 epochs while the baseline models are trained for 80-120 epochs depending on the model. Fine-tuning of ChemBERTaLM has taken approximately 6 hours in one Nvidia V100 GPU while training times for the baseline models were not reported. Although the models cannot be compared with respect to the exact training time without retraining, the dramatic difference in the number of training epochs suggests the utility of warm-start.

Table 4.1. Performance metrics for ChemBERTaLM and baseline models: fraction of valid compounds, fraction of compounds passing filters, internal diversity and novelty.

| **Model** | **Valid** | **Filters** | **IntDiv** | **IntDiv2** | **Novelty** |
|---|---|---|---|---|---|
| Train | 1 | 1 | 0.857 | 0.851 | 1 |
| AAE | 0.937 | 0.996 | 0.856 | 0.850 | 0.793 |
| CharRNN | 0.975 | 0.994 | 0.856 | 0.850 | 0.842 |
| VAE | 0.977 | **0.997** | 0.856 | 0.850 | 0.695 |
| LatentGAN | 0.897 | 0.973 | 0.857 | 0.850 | **0.949** |
| JTN-VAE | 1 | 0.978 | 0.851 | 0.845 | 0.914 |
| ChemBERTaLM | **0.991** | **0.997** | 0.855 | 0.849 | 0.844 |

Table 4.2. Performance metrics for ChemBERTaLM and baseline models on random test set (Test) and scaffold split test set (TestSF) from MOSES benchmark.

| **Model** | **Test** | | | | **TestSF** | | | |
|---|---|---|---|---|---|---|---|---|
| | **FCD (↓)** | **SNN (↑)** | **Frag (↑)** | **Scaf (↑)** | **FCD (↓)** | **SNN (↑)** | **Frag (↑)** | **Scaf (↑)** |
| Train | 0.008 | 0.642 | 1 | 0.991 | 0.476 | 0.586 | 0.999 | 1 |
| AAE | 0.556 | 0.608 | 0.991 | 0.902 | 1.057 | 0.568 | 0.990 | 0.079 |
| CharRNN | **0.073** | 0.601 | **1** | 0.924 | 0.520 | 0.565 | 0.998 | **0.110** |
| VAE | 0.099 | **0.626** | 0.999 | 0.939 | 0.567 | **0.578** | 0.998 | 0.059 |
| LatentGAN | 0.296 | 0.538 | 0.999 | 0.886 | 0.824 | 0.514 | 0.998 | 0.100 |
| JTN-VAE | 0.422 | 0.556 | 0.996 | 0.892 | 0.996 | 0.527 | 0.995 | 0.100 |
| ChemBERTaLM | 0.090 | 0.609 | **1** | 0.917 | **0.515** | 0.572 | 0.998 | 0.101 |

## 4.2. Target specific models

To investigate the effect of the warm start strategies, we trained the targeted generative models (i.e. EncDecBase, EncDecLM, and T5) on different data regimes (5%, 25%, and 100%). Then, for each protein in the test set, we generated 20 compounds from each model with two decoding methods, beam search and sampling.

### 4.2.1. Benchmarking metrics

To evaluate the performance of these models, we first computed a set of metrics assessing the feasibility of the generated compounds and their similarity to the compounds active towards the targets in our test set. The results are shown in Table 4.3.

Table 4.3. Performance metrics for target specific models on different percentages of data and decoding strategies.

| Decoding | Dataset | Model | % Valid | % Unique | % Novel | FCD | Scaf | SNN |
|---|---|---|---|---|---|---|---|---|
| BEAM SEARCH | 5% | EncDecLM | 0.968 | 0.686 | 0.987 | 17.012 | 0.050 | 0.574 |
| | | EncDecBase | 0.852 | 0.688 | 0.995 | 20.977 | 0.068 | 0.597 |
| | | T5 | 0.277 | 0.750 | 1.000 | 50.771 | 0.010 | 0.496 |
| | 25% | EncDecLM | 0.979 | 0.745 | 0.975 | 13.260 | 0.060 | 0.569 |
| | | EncDecBase | 0.959 | 0.738 | 0.980 | 12.653 | 0.068 | 0.572 |
| | | T5 | 0.801 | 0.811 | 0.998 | 25.260 | 0.056 | 0.547 |
| | 100% | EncDecLM | 0.984 | 0.795 | 0.965 | 9.454 | 0.090 | 0.560 |
| | | EncDecBase | 0.961 | 0.780 | 0.978 | 11.652 | 0.100 | 0.572 |
| | | T5 | 0.862 | 0.909 | 0.999 | 19.520 | 0.043 | 0.506 |
| SAMPLING | 5% | EncDecLM | 0.779 | 0.971 | 0.998 | 4.658 | 0.070 | 0.375 |
| | | EncDecBase | 0.628 | 0.940 | 0.998 | 6.691 | 0.062 | 0.391 |
| | | T5 | 0.127 | 1.000 | 1.000 | 11.152 | 0.017 | 0.313 |
| | 25% | EncDecLM | 0.860 | 0.956 | 0.995 | 4.791 | 0.078 | 0.406 |
| | | EncDecBase | 0.796 | 0.968 | 0.995 | 5.108 | 0.082 | 0.387 |
| | | T5 | 0.499 | 1.000 | 0.999 | 5.133 | 0.030 | 0.342 |
| | 100% | EncDecLM | 0.908 | 0.967 | 0.992 | 4.519 | 0.088 | 0.413 |
| | | EncDecBase | 0.840 | 0.986 | 0.996 | 4.366 | 0.085 | 0.389 |
| | | T5 | 0.664 | 1.000 | 1.000 | 4.688 | 0.032 | 0.341 |

The warm-started models performed fairly well even with 5% of the data and outperformed the T5 model trained from scratch. EncDecLM model performed the best or on par across metrics over different percentages of the data. However, the gain obtained by the pre-finetuning on molecule generation was limited especially in

terms of similarity metrics (i.e. FCD, Scaf, SNN). EncDecBase model showed better scores than the EncDecLM model with respect to the scaffold similarity (Scaf) and the nearest neighbor Tanimoto similarity (SNN) on 25% and 100% of the data. The metrics illustrate the novelty and precision of designed chemical compounds. This suggests that EncDecBase model generalizes better than EncDecLM model. We suppose that the pre-finetuning stage in EncDecLM might lead the model to forget previously learned knowledge.

We observed a dramatic improvement in the performance of the T5 model as the interactions are increased from 5% to 25%. This observation is not valid for the warm-started models. These models got slightly better scores with the increment of the data from 5% to 25%. With further increase of the interactions to 100%, all models achieved their best validity, uniqueness, and FCD scores. However, the scaffold similarity of the compounds generated by the T5 model with the beam search drops from 0.056 to 0.043 which might be an early sign of overfitting.

The lowest values of FCD scores were obtained with the compounds generated by sampling. This means that the sampling method produces more diverse compounds compared to the beam search. To investigate the diversity, we visualized the compounds with t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction technique [77] on the ECFP4 fingerprints. Figure 4.1 shows the mapping of the generated compounds with EncDecBase model on 100% of the data and randomly selected 5000 compounds from the test set in the molecular space. The visualization of the compounds generated by EncDecLM and T5 trained on 100% of the interactions are also depicted in Figure A.1 and A.2 respectively. The compounds generated by the beam search clustered on certain regions of the space similar to those from the test set while the ones generated with sampling scattered over the space.

To further investigate the performance of the models, we computed the evaluation metrics at the protein level. For each protein in the test set, we computed the FCD score between the compounds generated for this protein and the compounds active

towards this protein (i.e. those labeled as active). For comparison, we computed FCD scores for the proteins in the training set by randomly selecting 20 interacting compounds and computing FCD between the selected chemical compounds and the remaining binders of the target protein. Figure 4.2 shows the distribution of FCD scores across the percentages of the data and the decoding strategies. The results show that the warm-started models achieve lower FCDs than the T5 model on the target level, however, there is no significant improvement in the performance of these models with the increment of the data in terms of FCD. By contrast, there is a little but consistent decrease in FCD scores of the T5 model.

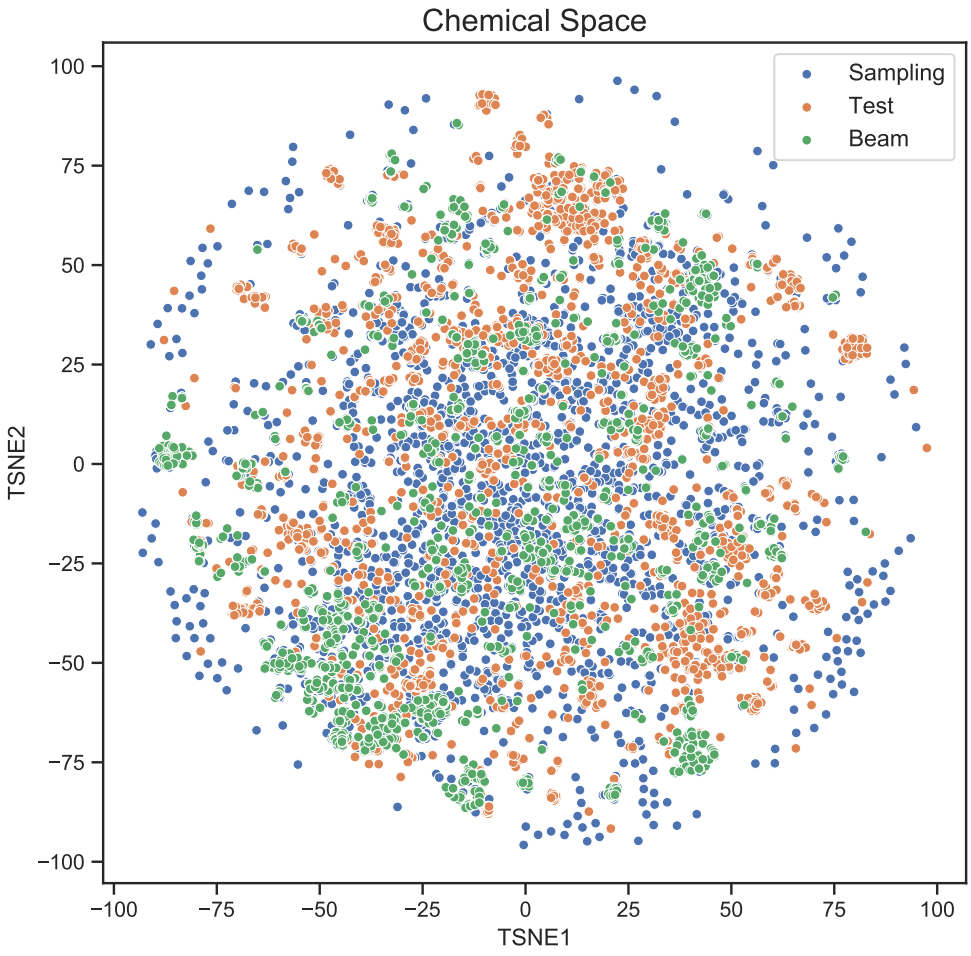


Figure 4.1. t-SNE plot of generated compounds with EncDecBase model trained on all interactions and 5000 randomly sampled compounds from the test set.

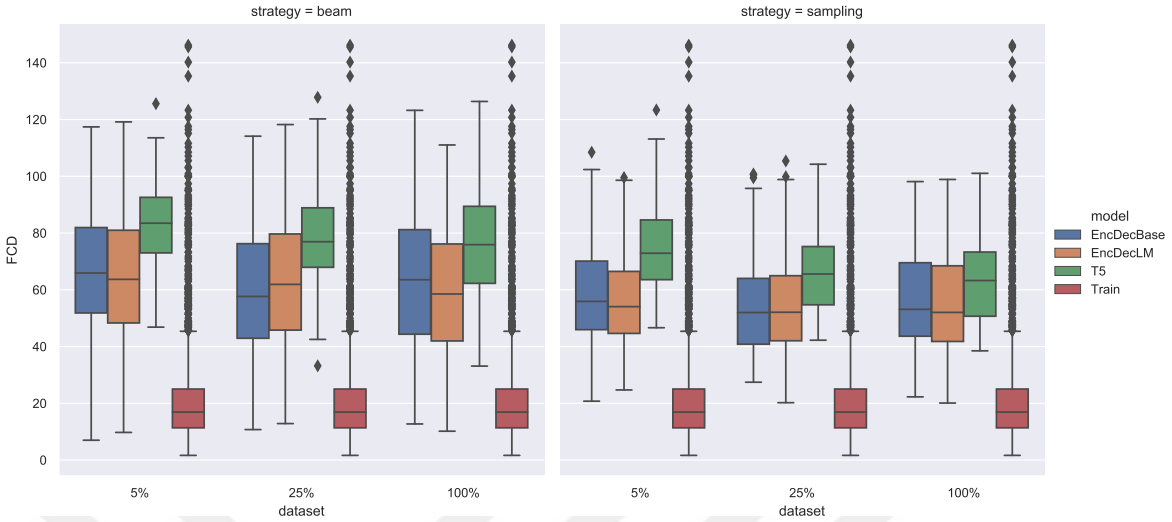


Figure 4.2. FCD score distribution of chemical compounds generated for test proteins across models, data levels and decoding strategies along with the distribution of the scores within interacting compounds of train proteins.

We also followed the same procedure to compute nearest neighbor Tanimoto similarity (SNN) scores for the proteins. The distributions of SNN scores are shown in Figure 4.3. SNN distributions also confirm that the warm started models perform better than T5 models across the data levels. Contrary to FCD scores, there is no significant improvement in the performance of T5 models with respect to SNN scores. Given that the FCD metric assesses the molecules in multiple aspects and SNN can be interpreted as precision, this result could imply that the properties of targeted chemical compounds among the generated ones by T5 model improve as the model trained with more data.

To compare the models on the protein level quantitatively, we computed Jensen-Shannon divergence (JSD) between the FCD score distribution for the test proteins with the compounds produced by the models and the FCD scores we computed for the train proteins. The results are shown in Table 4.4. We observed that the compounds decoded with the beam search obtained a lower JSD score for the warm started models while the compounds decoded with the sampling strategy result in lower JSD for the T5 models.

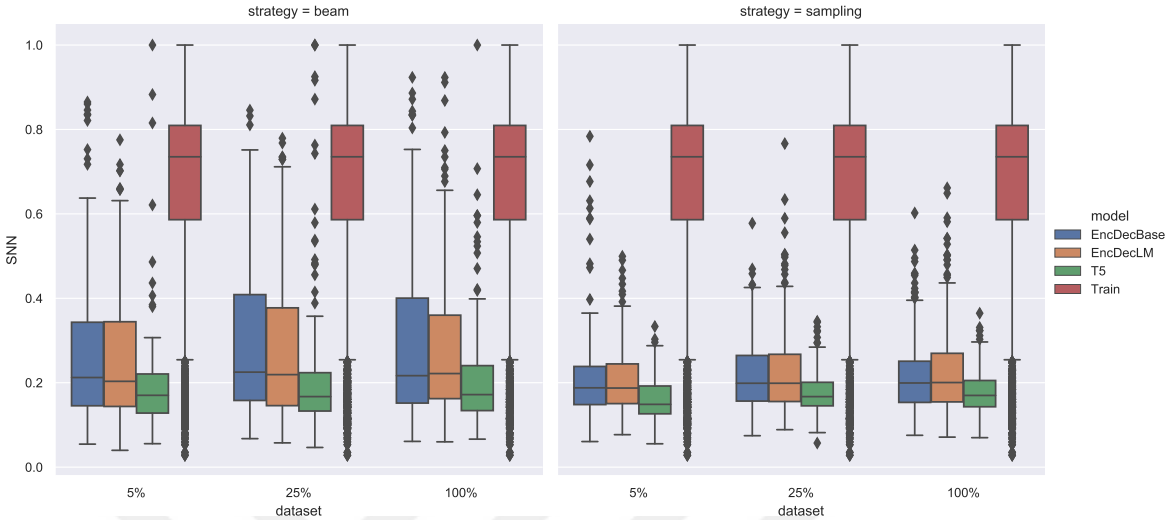


Figure 4.3. SNN distribution of chemical compounds generated for test proteins across models, data levels and decoding strategies along with the distribution of the scores within interacting compounds of train proteins.

Table 4.4. Jensen Shannon distance between FCD scores of the compounds generated for the test proteins and the scores of the compounds interacting with the train proteins.

| **Dataset** | **Model** | **Beam** | **Sampling** |
|---|---|---|---|
| 5% | EncDecBase | 0.605 | 0.627 |
| | EncDecLM | 0.601 | 0.637 |
| | T5 | 0.728 | 0.723 |
| 25% | EncDecBase | 0.549 | 0.606 |
| | EncDecLM | 0.582 | 0.596 |
| | T5 | 0.706 | 0.700 |
| 100% | EncDecBase | **0.546** | 0.592 |
| | EncDecLM | 0.557 | **0.573** |
| | T5 | 0.694 | 0.683 |

### 4.2.2. Docking

Next, we performed docking to assess if generated chemical compounds are active against the protein of interest. The target proteins were selected randomly from the test proteins which have at least 50 interacting compounds and 3D structure in complex with a ligand. Given that docking is a computationally costly process, we limited the number of proteins for docking to 12. For each protein, we selected one structure from PDB [41]. The PDB codes for selected proteins are as follows: 1ERE, 2WO6, 4B6L, 4I23, 5K0K, 5TUY, 5V1B, 5XY1, 6LVL, 6WJ5, 6Z1Q, 2PJL.

To test whether generated compounds have binding affinity to the target protein of interest (i.e. target specificity), we compiled three groups of compounds for each protein: the compounds having activity towards the target protein, the compounds randomly selected from BindingDB, the generated molecules for that protein. Due to technical limitations, the maximum size of these groups is set to 100. If a group contains more compounds, then 100 compounds are randomly chosen.

The ligands in each group were docked into corresponding protein structures with GNINA [42] which is a molecular docking tool based on deep learning. To achieve this, we first extracted bound ligands from PDB structures and saved the ligand and protein separately using PyMol [78]. The extracted ligands were used to define the binding site of the target for docking. To dock a compound to a target, we also need a 3D structure of that compound. Therefore, we used RDKit [74] to generate conformers for the chemical compounds. Docking was performed with the default parameters of GNINA by specifying extracted ligands to define binding sites. For docking of each compound into its target, the tool generates multiple poses with two scores: the one used to rank the poses of the ligand (i.e. CNNscore) and the one estimating the affinity of the docked complex (i.e. CNNaffinity). Since we are interested in the binding affinity of compounds against targets, we selected the pose with the highest affinity for each ligand-protein complex.

Next, we first assessed the discrimination ability of this docking tool by comparing the predicted affinity scores of compounds active towards a target with the scores of randomly sampled compounds for each structure. We chose Receiving Operating Characteristic (ROC) curve and Area Under the ROC Curve (AUC) as evaluation metrics considering the active compounds as positive examples and randomly selected compounds as negative examples. These metrics measure whether this tool is able to distinguish between positive and negative classes. We also performed Mann-Whitney's U test to check the discrimination between the compounds is significant or not. ROC curves for all structures are plotted in Figure 4.4. AUC scores and p values of Mann-Whitney's U test are given in parenthesis. For 11 out of 12 proteins, the tool was able to discriminate between the active compounds and random ones. Because for the structure with PDB code 2PJL the active compounds and randomly selected ones cannot be distinguished, we excluded it from the rest of the experiments.

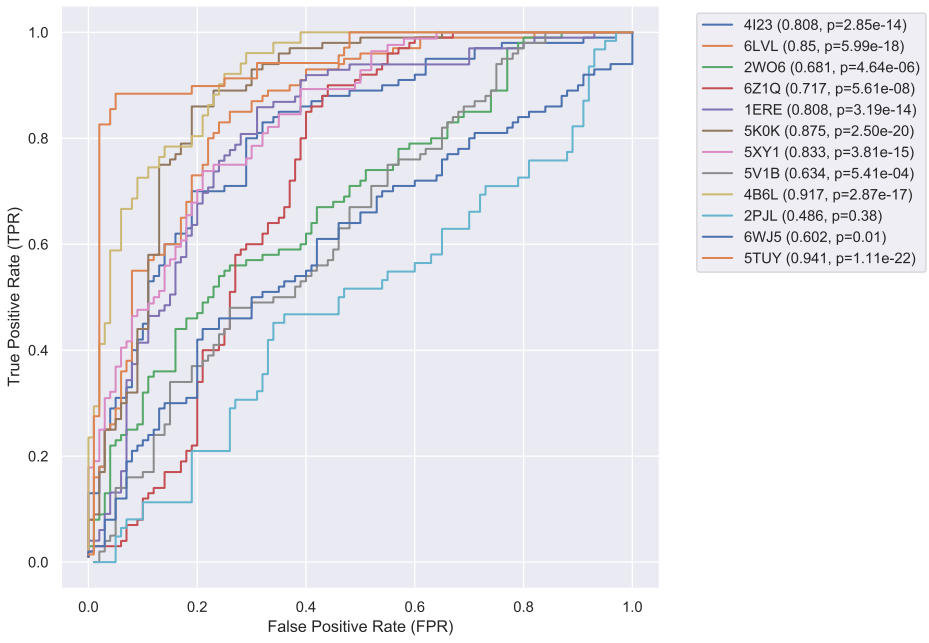


Figure 4.4. ROC Curves for the comparison of active compounds to random ones for each protein structure. AUC scores and p values (Mann-Whitney's U test) are reported in parenthesis respectively.

Once the discrimination ability of the docking tool is validated, we next investigated whether generated compounds can bind to the intended targets. To achieve this, we compared these compounds with randomly selected compounds and the active compounds of the protein of interest. If two sets of compounds cannot be distinguished from one another, this suggests that the binding affinity of these compounds towards the target is similar. On the other hand, the groups which can be discriminated against are more or less likely to bind to the target than the other depending on the AUC score. A high AUC score ($> 0.6$) indicates that the compounds considered as the positive class show higher scores than those considered as the negative class. Likewise, a low AUC score ($< 0.4$) shows that the tool is more likely to classify the compounds chosen as negative instances as binders.

For each target protein, we assessed the affinity of the compounds generated with both decoding methods by the models trained on 25% and 100% of the interactions since the scores of these models are close in terms of benchmark metrics. We first compared the generated set with the random set. In this setting, the generated set is considered as the positive class while the random set forms the negative class. For each structure-compound set, ROC curves and AUC scores were computed and Mann-Whitney's U test was performed similarly to the previous experiment. Given that the number of proteins evaluated is high, here ROC curves for only two protein structures are shown in Figure 4.5 and 4.6 for the sake of space. ROC curves for the remaining proteins can be found in Section B. AUC scores and p values (Mann-Whitney's U test) are reported in parenthesis.

Mitogen-activated protein kinase kinase kinase 14 (MAP3K14) is one of the target proteins and plays an essential role in the activation of NF-kappa-B signaling. For this reason, it is also known as NF-B-inducing kinase (NIK). This pathway mediated by NIK is involved in severe immune diseases and new blood vessel formation in cancer [79]. Hence, this target is a potential target for treating such diseases. Figure 4.5 shows the comparison of the generated compounds by the models with the random set (on the left) and the active molecules (on the right) for this target. In comparison between the

generated set and random set, high AUC scores were obtained by all models except the T5 model trained on 100%. This suggests that these models are able to generate compounds with higher affinity than the random molecules. The group of generated molecules by these models also either cannot be distinguished from the group of actives (p-value $>$ 0.05) or obtain higher affinity scores than the active group (p-value $<$ 0.05 and AUC $>$ 0.6). One interesting observation is that the ligands generated with sampling tend to have higher affinity scores than the ones generated with beam search for this target.

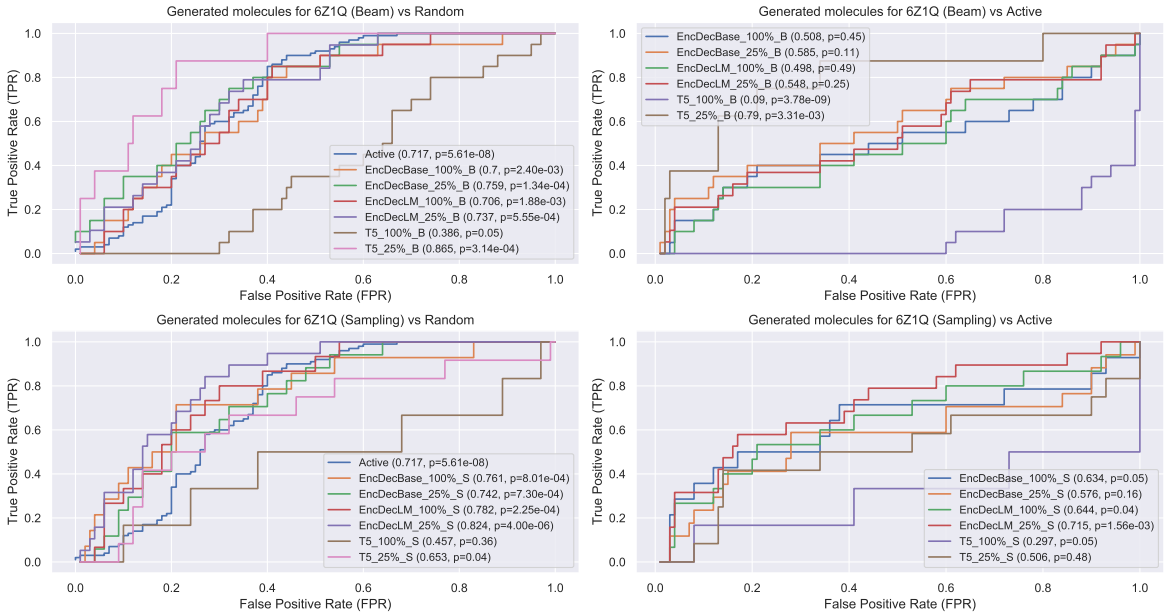


Figure 4.5. ROC Curves for the comparison of the generated compounds for MAP3K14 with random compounds and active compounds. The labels indicate the model type, the data level the model trained on and the decoding strategy.

Polo-like kinase 3 (PLK3) is another target for which we investigated the generated molecules. PLK3 is one of the key regulators of cell cycle progression. This target has been identified as a tumor suppressor for certain cancer types, however, recent studies have shown that PLK3 plays different roles in cancer types and identified this protein as a potential target in colorectal cancer [80]. The computed scores for the compounds targeting this protein are shown in Figure 4.6. It can be seen that the compounds generated by warm-started models differ significantly from the random set.

By contrast, T5 models which are trained from scratch failed to produce such compounds. Although the compounds obtained higher affinity scores compared to the random set for all warm-started models, only the ones generated by EncDecLM models with beam decoding cannot be discriminated from the active compounds of the target. EncDecBase model trained on 100% is the only model generating the compounds which have significantly higher affinity against the target of interest than both the random and the known active set. This result is achieved by the compounds decoded with beam search. For all warm started models, beam search generated compounds are more likely to bind to this target.

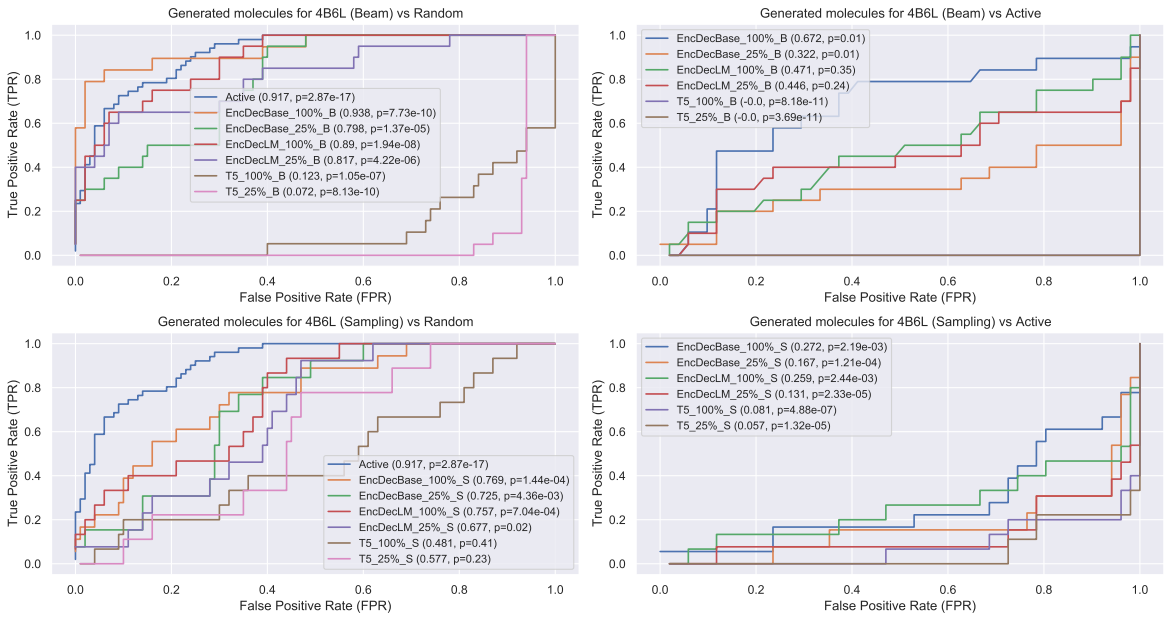


Figure 4.6. ROC Curves for the comparison of the generated compounds for PLK3 with random compounds and active compounds.The labels indicate the model type, the data level the model trained on and the decoding strategy.

To determine the overall performance of these models, we computed the number of proteins for which the model can generate compounds that can be distinguished from randomly selected compounds and bind to the corresponding target as likely as or more likely than the known binders of that target. The results are reported in Table 4.2.2. The highest number of proteins (11), for which generated compounds are more likely to bind to a target of interest than randomly selected compounds were obtained with

the compounds generated by the EncDecBase model trained on 100% and with beam decoding. This result is consistent with the target level evaluation based on FCD scores and suggests that this model can generate chemical compounds with binding affinity to the target of interest.

Table 4.5. Number of proteins in which generated compounds can be distinguished from random compounds for each model and decoding strategy.

| Decoding | Dataset | Model | Generated vs Random | Generated vs Active |
|---|---|---|---|---|
| Beam | 100% | EncDecBase | 11 | 8 |
| | | EncDecLM | 8 | 6 |
| | | T5 | 7 | 5 |
| | 25% | EncDecBase | 7 | 6 |
| | | EncDecLM | 7 | 6 |
| | | T5 | 4 | 3 |
| Sampling | 100% | EncDecBase | 8 | 4 |
| | | EncDecLM | 8 | 4 |
| | | T5 | 2 | 1 |
| | 25% | EncDecBase | 6 | 2 |
| | | EncDecLM | 6 | 2 |
| | | T5 | 2 | 2 |

EncDecBase model trained on all interactions performed the best and could generate compounds with binding affinity to a target of interest when decoded with beam search. However, the generated compounds must also be specific to the target to avoid undesired side effects caused by binding to others. To assess the target specificity of this model, we selected 100 compounds generated for others for each target and docked this set of molecules to the target protein. Then, we compared the binding affinity of this set to the following sets we compiled earlier: the compounds generated for the target of interest and the compounds active towards the target. The results are summarized in Table 4.6. For comparison, we also included the results of the generated compounds for the targets compared to the random molecules and the active compounds.

We observed that high AUC values were obtained when comparing the active compounds versus the compounds generated for others. This indicates that the docking tool is more likely to classify the compounds targeting others as nonbinders for the target of interest. By contrast, the generated compounds for only three targets (1ERE, 5TUY, 5XY1) were more likely to be classified as nonbinders. Additionally, the comparison between the compounds generated for the target of interest and those generated for others shows that these sets of compounds are significantly different from each other for all targets except 1ERE and 6Z1Q. Taken together, these results suggest that this model can generate target specific compounds for the majority of the targets.

Table 4.6. Comparison of activity against each target between the active compounds, the generated ones for the target and the generated ones for the others.

| **Target** | **Active vs Generated** | | **Active vs Others** | | **Generated vs Others** | |
|---|---|---|---|---|---|---|
| | **p value** | **AUC** | **p value** | **AUC** | **p value** | **AUC** |
| **1ERE** | 1.74E-05 | 0.79 | 1.73E-06 | 0.69 | 0.418 | 0.52 |
| **2WO6** | 0.013 | 0.33 | 0.007 | 0.6 | 0.001 | 0.74 |
| **4B6L** | 0.014 | 0.33 | 1.98E-13 | 0.87 | 1.81E-08 | 0.9 |
| **4I23** | 0.027 | 0.36 | 4.46E-08 | 0.72 | 2.11E-06 | 0.83 |
| **5K0K** | 0.332 | 0.47 | 2.84E-09 | 0.74 | 9.28E-05 | 0.77 |
| **5TUY** | 4.90E-06 | 0.83 | 2.77E-14 | 0.85 | 0.001 | 0.73 |
| **5V1B** | 2.28E-06 | 0.17 | 0.035 | 0.58 | 4.76E-07 | 0.85 |
| **5XY1** | 0.018 | 0.65 | 1.56E-11 | 0.79 | 0.001 | 0.72 |
| **6LVL** | 0.389 | 0.48 | 4.50E-13 | 0.8 | 3.95E-06 | 0.82 |
| **6WJ5** | 4.58E-05 | 0.22 | 0.012 | 0.59 | 2.69E-06 | 0.82 |
| **6Z1Q** | 0.454 | 0.49 | 0.053 | 0.57 | *0.175* | 0.57 |

### 4.3. Synthesizability

Apart from having binding affinity to a target, drug candidates must satisfy many other constraints such as being synthetically feasible. It is shown that generative models often generate compounds hard to synthesize. Therefore, we assessed synthesizability of the generated compounds by the best model (i.e. EncDecBase trained on 100%).

We estimated the synthetic accessibility of the compounds for the eight targets where the docking tool is more likely to classify the compounds as binders using Molecule.One API [81]. Figure 4.7 shows the distribution of estimated scores across targets. The score ranges between 1 and 10, estimating the cost of synthesizing the molecule [82]. Although the scores are high for the half of the targets on average, the model is able to generate synthetically feasible compounds for all targets except one (5V1B). Molecule.One API also provides retrosynthesis planning of these compounds. We included the synthesis planning of the compound with lowest synthetic accessibility score for each target since it might help the researchers studying such targets. The path for the compound targeting PLK3 protein we mentioned before is depicted in Figure 4.8 while the others can be found in Appendix C.

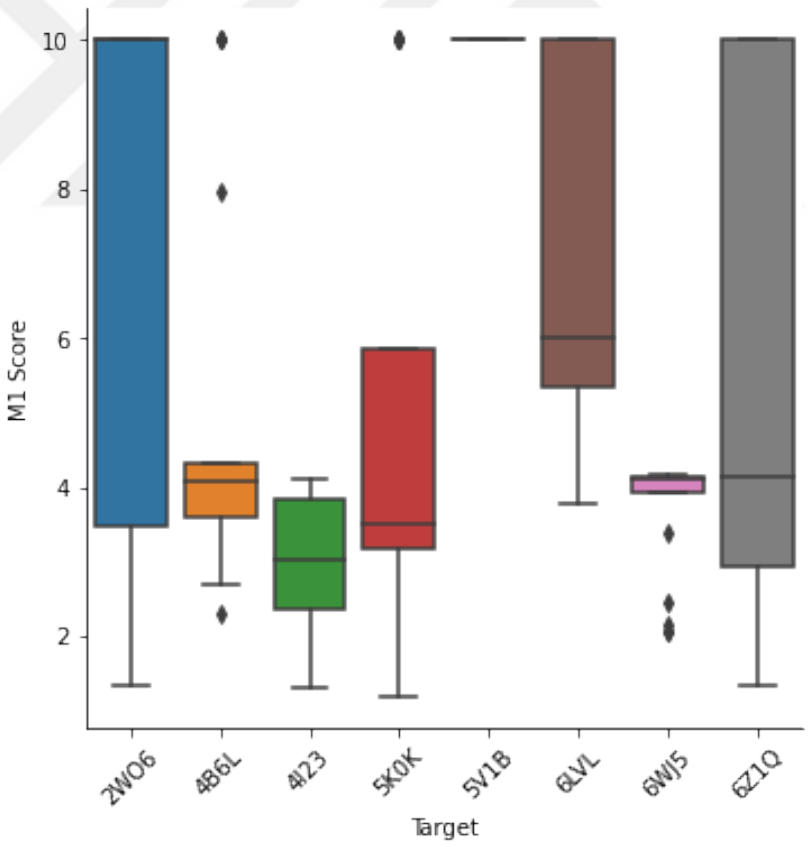


Figure 4.7. Synthesizability scores of generated compounds by the best performing model across targets.

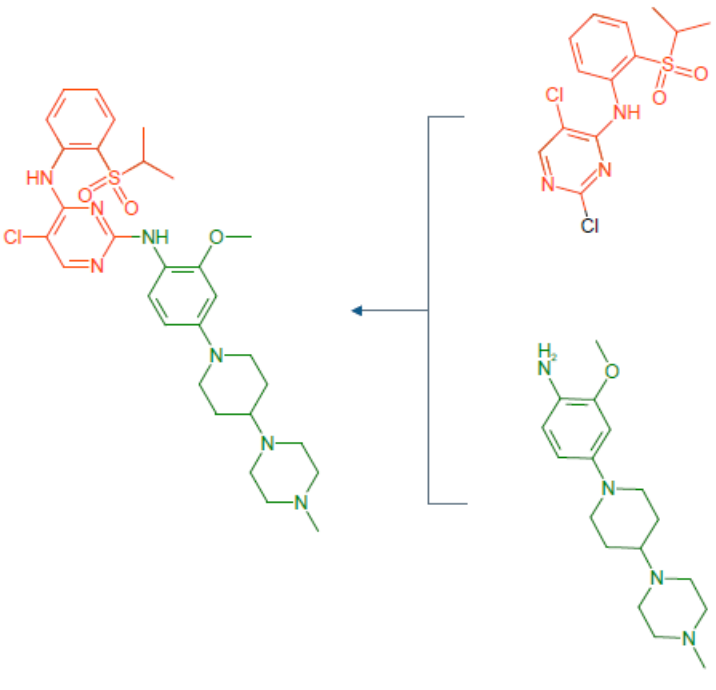


Figure 4.8. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 4B6L.

# 5. DISCUSSION

Our findings suggest that the proposed warm start strategies to initialize models outperform the T5 model trained from scratch across data regimes and decoding methods. The proposed method might be beneficial for other tasks in cheminformatics where labeled data are limited. Even when large amount of data are available, it might reduce computational time. However, we should also note that pretrained models used to initialize models require more space. The results demonstrate that the model warm started with the one stage strategy is able to generate target specific compounds for most of the set of novel proteins. Thus, it might be practical at the beginning of the drug design where there is little information regarding the target of interest.

We framed targeted drug design as a translation problem from protein "language" to chemical "language". Considering the similarities between biochemical languages and human languages and the advantages regarding performance, we treated the tokens identified by subword segmentation algorithm as "words" of these languages. However, given that there are no clear word boundaries in the protein language, and protein sequences are in general much longer than sentences in human language, this assumption requires further investigation. These so-called words can be analyzed in terms of the statistical similarity to words in human language and compared with known functional/structural units of proteins and chemicals.

Despite the potential benefits, the proposed model has certain limitations. The compounds generated by beam search are predicted to be able to bind to the protein of interest but have low diversity. The deterministic nature of the model combined with the variability of protein interactions hinders further generalizability of the model. To overcome this issue, one may incorporate stochastic latent variables into the model to improve the capability to learn variability of interactions [83, 84]. Another direction to improve compound diversity is to use ancestral sampling methods which have been proposed recently and have been shown to generate high diversity samples in [85–87].

Another concerning issue regarding the model is the synthetic accessibility. Like many other deep generative models, our model is in general not able to generate compounds that are easy to synthesize. The studies addressing this limitation either enforce synthetic feasibility explicitly by employing reaction or synthesis based approaches [88] or implicitly through inductive biases [89]. Such methods can be adopted by extending our model with a component ensuring synthesizability of molecules explicitly or indirectly biasing the model with iterative mutation of compounds using interchangeable fragments extracted from synthetically feasible compounds.

# 6. CONCLUSION

In this thesis, we investigated warm-starting models with pretrained models for targeted drug design task. We compared two warm-start strategies: one stage strategy where the model is initialized with pretrained checkpoints and trained on targeted molecule generation (EncDecBase models) and two stage strategy where the model is first finetuned on molecular generation and then trained on targeted drug design (EncDecLM models). To generate molecules, we employed beam search and sampling. We evaluated the generated molecules with a set of metrics assessing the quality and diversity, and docking. The results showed the efficacy of warm-starting approach. The warm-started models obtained significantly better scores than the baseline model (i.e. T5) across data regimes and decoding methods. The warm-started models performed on par with each other with respect to the metrics. However, based on docking results, the one stage warm-start strategy was able to generate molecules likely to bind to the target of interest for more proteins, indicating that the model has better generalizability.

As future work, we plan to investigate adding stochastic latent variables into our models to increase diversity of compounds similar to Variational Transformer models [83,84]. Additionally, to mitigate the synthesizability issue which limits the practicality of the model, we will explore applicability of the approaches ensuring feasibility by joint optimization of generation and synthesis [88].

# REFERENCES


1. Polishchuk, P. G., 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*, Vol. 27, No. 8, pp. 675–679, 2013.

2. Peón, A., S. Naulaerts and P. J. Ballester, "Predicting the reliability of drug-target interaction predictions with maximum coverage of target space", *Scientific Reports*, Vol. 7, No. 1, pp. 1–11, 2017.

3. Kramer, J. A., J. E. Sagartz and D. L. Morris, "The application of discovery toxicology and pathology towards the design of safer pharmaceutical lead candidates", *Nature Reviews Drug Discovery*, Vol. 6, No. 8, pp. 636–649, 2007.

4. Mouchlis, V. D., A. Afantitis, A. Serra, M. Fratello, A. G. Papadiamantis, V. Aidinis, I. Lynch, D. Greco and G. Melagraki, "Advances in de novo drug design: From conventional to machine learning methods", *International Journal of Molecular Sciences*, Vol. 22, No. 4, pp. 1–22, 2021.

5. Skinnider, M. A., R. G. Stacey, D. S. Wishart and L. J. Foster, "Deep generative models enable navigation in sparsely populated chemical space", *ChemRxiv*, 2021.

6. Aumentado-Armstrong, T., "Latent molecular optimization for targeted therapeutic design", *arXiv preprint arXiv:1809.02032*, 2018.

7. Skalic, M., 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*, Vol. 16, No. 10, pp. 4282–4291.

8. Moret, M., L. Friedrich, F. Grisoni, D. Merk and G. Schneider, "Generative molecular design in low data regimes", *Nature Machine Intelligence*, Vol. 2, No. 3, pp. 171–180, 2020.

9. Segler, M. H., T. Kogej, C. Tyrchan and M. P. Waller, "Generating focused molecule libraries for drug discovery with recurrent neural networks", *ACS Central Science*, Vol. 4, No. 1, pp. 120–131, 2018.

10. Zhavoronkov, A., 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, Y. Volkov, A. Zholus, R. R. Shayakhmetov, A. Zhebrak, L. I. Minaeva, B. A. Zagribelnyy, L. H. Lee, R. Soll, D. Madge, L. Xing, T. Guo and A. Aspuru-Guzik, "Deep learning enables rapid identification of potent DDR1 kinase inhibitors", *Nature Biotechnology*, Vol. 37, No. 9, pp. 1038–1040, 2019.

11. Born, J., M. Manica, J. Cadow, G. Markert, N. A. Mill, M. Filipavicius, N. Janakarajan, A. Cardinale, T. Laino and M. R. Martínez, "Data-driven molecular design for discovery and synthesis of novel ligands: A case study on SARS-CoV-2", *Machine Learning: Science and Technology*, Vol. 2, No. 2, p. 25024, 2021.

12. Chenthamarakshan, V., P. Das, I. Padhi, H. Strobelt, K. W. Lim, B. Hoover, S. C. Hoffman and A. Mojsilovic, "Target-specific and selective drug design for covid-19 using deep generative models", *arXiv preprint arXiv:2004.01215*, 2020.

13. Popova, M., O. Isayev and A. Tropsha, "Deep reinforcement learning for de novo drug design", *Science Advances*, Vol. 4, No. 7, p. eaa7885, 2018.

14. Burley, S. K., C. Bhikadiya, C. Bi, S. Bittrich, L. Chen, G. V. Crichlow, C. H. Christie, K. Dalenberg, L. Di Costanzo, J. M. Duarte *et al.*, "RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D437–D451, 2021.

15. "UniProt: the universal protein knowledgebase in 2021", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D480–D489, 2021.

16. Woźniak, M., A. Wołos, U. Modrzyk, R. L. Górski, J. Winkowski, M. Bajczyk, S. Szymkuć, B. A. Grzybowski and M. Eder, "Linguistic measures of chemical diversity and the "keywords" of molecular collections", *Scientific Reports*, Vol. 8, No. 1, p. 7598, 2018.

17. Özçelik, R., H. Öztürk, A. Özgür and E. Ozkirimli, "ChemBoost: A chemical language based approach for protein–ligand binding affinity prediction", *Molecular Informatics*, Vol. 40, No. 5, p. 2000212, 2021.

18. Chithrananda, S., G. Grand and B. Ramsundar, "ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction", *arXiv preprint arXiv:2010.09885*, 2020.

19. Filipavicius, M., M. Manica, J. Cadow and M. R. Martinez, "Pre-training protein language models with label-agnostic binding pairs enhances performance in downstream tasks", *arXiv preprint arXiv:2012.03084*.

20. Rives, A., 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*, Vol. 118, No. 15, 2021.

21. Elnaggar, A., M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger *et al.*, "ProtTrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing", *arXiv preprint arXiv:2007.06225*, 2020.

22. Li, X. and D. Fourches, "SMILES Pair Encoding: A data-driven substructure tokenization algorithm for deep learning", *Journal of Chemical Information and Modeling*, Vol. 61, No. 4, pp. 1560–1569, 2021.

23. Asgari, E., A. C. McHardy and M. R. K. Mofrad, "Probabilistic variable-length

segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX)", *Scientific Reports*, Vol. 9, No. 1, p. 3577, Mar. 2019.

24. Szklarczyk, D., A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. T. Doncheva, J. H. Morris, P. Bork, L. J. Jensen and C. Von Mering, "STRING v11: Protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets", *Nucleic Acids Research*, Vol. 47, No. D1, pp. D607–D613, 2019.

25. Sennrich, R., B. Haddow and A. Birch, "Neural machine translation of rare words with subword units", *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, Vol. 3, pp. 1715–1725, 2015.

26. "PubChem 2019 update: Improved access to chemical data", *Nucleic Acids Research*, Vol. 47, No. D1, pp. D1102–D1109, 2019.

27. Gilson, M. K., 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*, Vol. 44, No. D1, pp. D1045–D1053, 2016.

28. Polykovskiy, D., A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, A. Kadurin, S. Johansson, H. Chen, S. Nikolenko, A. Aspuru-Guzik and A. Zhavoronkov, "Molecular Sets (MOSES): A benchmarking platform for molecular generation models", *Frontiers in Pharmacology*, Vol. 11, pp. 1–19, 2020.

29. Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer", *Journal of Machine Learning Research*, Vol. 21, pp. 1–67, 2019.

30. Grechishnikova, D., "Transformer neural network for protein-specific de novo drug generation as a machine translation problem", *Scientific Reports*, Vol. 11, No. 1, pp. 1–13, 2021.

31. Paul, S. M., D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg and A. L. Schacht, "How to improve R&D productivity: the pharmaceutical industry's grand challenge", *Nature Reviews Drug discovery*, Vol. 9, No. 3, pp. 203–214, 2010.

32. Govardhanagiri, S., S. Bethi and G. P. Nagaraju, "Small molecules and pancreatic cancer trials and troubles", *Breaking Tolerance to Pancreatic Cancer Unresponsiveness to Chemotherapy*, pp. 117–131, Elsevier, 2019.

33. Mahmood, O., E. Mansimov, R. Bonneau and K. Cho, "Masked graph modeling for molecule generation", *Nature Communications*, Vol. 12, No. 1, pp. 1–12, 2021.

34. Samanta, B., A. De, G. Jana, P. K. Chattaraj, N. Ganguly and M. Gomez-Rodriguez, "NeVAE: A deep generative model for molecular graphs", *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 1110–1117, 2018.

35. Li, Y., L. Zhang and Z. Liu, "Multi-objective de novo drug design with conditional graph generative model", *Journal of Cheminformatics*, Vol. 10, No. 1, p. 33, 2018.

36. Li, Y., O. Vinyals, C. Dyer, R. Pascanu and P. Battaglia, "Learning deep generative models of graphs", *arXiv preprint arXiv:1803.03324*, 2018.

37. Rogers, D. and M. Hahn, "Extended-connectivity fingerprints", *Journal of Chemical Information and Modeling*, Vol. 50, No. 5, pp. 742–754, 2010.

38. Rost, B., "Protein structure prediction in 1D, 2D, and 3D", *Encyclopedia of Computational Chemistry*, John Wiley  Sons, Ltd, 2002.

39. Needleman, S. B. and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology*, Vol. 48, No. 3, pp. 443–453, 1970.

40. Mistry, J., S. Chuguransky, L. Williams, M. Qureshi, G. A. Salazar, E. L. Sonnhammer, S. C. Tosatto, L. Paladin, S. Raj, L. J. Richardson, R. D. Finn and A. Bateman, "Pfam: The protein families database in 2021", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D412–D419, 2021.

41. Burley, S. K., H. M. Berman, C. Bhikadiya, C. Bi, L. Chen, L. Di Costanzo, C. Christie, J. M. Duarte, S. Dutta, Z. Feng, S. Ghosh, D. S. Goodsell, R. K. Green, V. Guranovic, D. Guzenko, B. P. Hudson, Y. Liang, R. Lowe, E. Peisach, I. Periskova, C. Randle, A. Rose, M. Sekharan, C. Shao, Y. P. Tao, Y. Valasatava, M. Voigt, J. Westbrook, J. Young, C. Zardecki, M. Zhuravleva, G. Kurisu, H. Nakamura, Y. Kengaku, H. Cho, J. Sato, J. Y. Kim, Y. Ikegawa, A. Nakagawa, R. Yamashita, T. Kudou, G. J. Bekker, H. Suzuki, T. Iwata, M. Yokochi, N. Kobayashi, T. Fujiwara, S. Velankar, G. J. Kleywegt, S. Anyango, D. R. Armstrong, J. M. Berrisford, M. J. Conroy, J. M. Dana, M. Deshpande, P. Gane, R. Gáborová, D. Gupta, A. Gutmanas, J. Koča, L. Mak, S. Mir, A. Mukhopadhyay, N. Nadzirin, S. Nair, A. Patwardhan, T. Paysan-Lafosse, L. Pravda, O. Salih, D. Sehnal, M. Varadi, R. Vařeková, J. L. Markley, J. C. Hoch, P. R. Romero, K. Baskaran, D. Maziuk, E. L. Ulrich, J. R. Wedell, H. Yao, M. Livny and Y. E. Ioannidis, "Protein Data Bank: The single global archive for 3D macromolecular structure data", *Nucleic Acids Research*, Vol. 47, No. D1, pp. D520–D528, 2019.

42. McNutt, A. T., P. Francoeur, R. Aggarwal, T. Masuda, R. Meli, M. Ragoza, J. Sunseri and D. R. Koes, "GNINA 1.0: molecular docking with deep learning", *Journal of Cheminformatics*, Vol. 13, No. 1, pp. 1–20, 2021.

43. Koes, D. R., M. P. Baumgartner and C. J. Camacho, "Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise", *Journal of Chemical Information and Modeling*, Vol. 53, No. 8, pp. 1893–1904, 2013.

44. Trott, O. and A. J. Olson, "AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading", *Journal of Computational Chemistry*, Vol. 31, No. 2, pp. NA–NA, 2009.

45. Gage, P., "A new algorithm for data compression", *C Users Journal*, Vol. 12, No. 2, pp. 23–38, 1994.

46. Kawano, K., S. Koide and C. Imamura, "Seq2seq fingerprint with byte-pair encoding for predicting changes in protein stability upon single point mutation", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 17, No. 5, pp. 1762–1772, 2019.

47. Asgari, E., A. C. McHardy and M. R. Mofrad, "Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX)", *Scientific Reports*, Vol. 9, No. 1, pp. 1–16, 2019.

48. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need", *Advances in Neural Information Processing Systems*, Vol. 2017-Decem, pp. 5999–6009, 2017.

49. Holtzman, A., J. Buys, L. Du, M. Forbes and Y. Choi, "The curious case of neural text degeneration", *CEUR Workshop Proceedings*, Vol. 2540, 2019.

50. Devlin, J., M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, Vol. 1, pp. 4171–4186, Association for Computational Linguistics (ACL), 2019.

51. Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach", *arXiv preprint arXiv:1907.11692*, 2019.

52. Weininger, D., "SMILES, a chemical language and information System: 1: Introduction to methodology and encoding rules", *Journal of Chemical Information and Computer Sciences*, Vol. 28, No. 1, pp. 31–36, 1988.

53. Gupta, A., A. T. Müller, B. J. Huisman, J. A. Fuchs, P. Schneider and G. Schneider, "Generative recurrent networks for de novo drug design", *Molecular Informatics*, Vol. 37, No. 1, 2018.

54. Mercado, R., T. Rastemo, E. Lindelof, G. Klambauer, O. Engkvist, H. Chen and E. J. Bjerrum, "Graph networks for molecular design", *Machine Learning: Science and Technology*, Vol. 2, No. 2, 2021.

55. Simonovsky, M. and N. Komodakis, "GraphVAE: Towards generation of small graphs using variational autoencoders", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 11139 LNCS, pp. 412–422, Springer Verlag, 2018.

56. Popova, M., O. Isayev and A. Tropsha, "Deep reinforcement learning for de novo drug design", *Science Advances*, Vol. 4, No. 7, p. eaap7885, 2018.

57. Zhou, Z., S. Kearnes, L. Li, R. N. Zare and P. Riley, "Optimization of molecules via deep reinforcement learning", *Scientific Reports*, Vol. 9, No. 1, pp. 1–10, 2019.

58. You, J., B. Liu, R. Ying, V. Pande and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation", *Advances in Neural Information Processing Systems*, Vol. 2018-Decem, pp. 6410–6421, 2018.

59. Gómez-Bombarelli, R., J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules", *ACS Central Science*, Vol. 4, No. 2, pp. 268–276, 2018.

60. Renz, P., D. Van Rompaey, J. K. Wegner, S. Hochreiter and G. Klambauer, "On failure modes in molecule generation and optimization", *Drug Discovery Today: Technologies*, 2020.

61. Boitreaud, J., V. Mallet, C. Oliver and J. Waldispuhl, "OptiMol: optimization of binding affinities in chemical space for drug discovery", *Journal of Chemical Information and Modeling*, Vol. 60, No. 12, pp. 5658–5666, 2020.

62. Dong, L., N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation", *Advances in Neural Information Processing Systems*, Vol. 32.

63. Song, K., X. Tan, T. Qin, J. Lu and T.-Y. Liu, "MASS: Masked sequence to sequence pre-training for language generation", *36th International Conference on Machine Learning, ICML 2019*, Vol. 2019-June, pp. 10384–10394, 2019.

64. Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding", *Advances in Neural Information Processing Systems*, Vol. 32.

65. Rao, R., N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel and Y. S. Song, "Evaluating protein transfer learning with TAPE", *Advances in Neural Information Processing Systems*, Vol. 32, p. 9689, 2019.

66. Vig, J., A. Madani, L. R. Varshney, C. Xiong, R. Socher and N. F. Rajani, "Bertology meets biology: Interpreting attention in protein language models", *arXiv preprint arXiv:2006.15222*, 2020.

67. Jansson-Löfmark, R., S. Hjorth and J. Gabrielsson, "Does in vitro potency predict clinically efficacious concentrations?", *Clinical Pharmacology & Therapeutics*, Vol. 108, No. 2, pp. 298–305, 2020.

68. Gao, K. Y., A. Fokoue, H. Luo, A. Iyengar, S. Dey and P. Zhang, "Interpretable

drug target prediction using deep neural representation", *IJCAI*, pp. 3371–3377, 2018.

69. Cock, P. J. A., T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski and M. J. L. de Hoon, "Biopython: freely available Python tools for computational molecular biology and bioinformatics", *Bioinformatics*, Vol. 25, No. 11, pp. 1422–1423, 03 2009.

70. Rice, P., I. Longden and A. Bleasby, "EMBOSS: the European molecular biology open software suite", *Trends in Genetics*, Vol. 16, No. 6, pp. 276–277, 2000.

71. Sterling, T. and J. J. Irwin, "ZINC 15–ligand discovery for everyone", *Journal of Chemical Information and Modeling*, Vol. 55, No. 11, pp. 2324–2337, 2015.

72. Fabian, B., 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.

73. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest and A. Rush, "Transformers: State-of-the-Art natural language processing", pp. 38–45, Association for Computational Linguistics (ACL).

74. Landrum, G. *et al.*, "RDKit: Open-source cheminformatics", `https://www.rdkit.org/`.

75. Preuer, K., 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*, Vol. 58, No. 9, pp. 1736–1741, 2018.

76. Bemis, G. W. and M. A. Murcko, "The properties of known drugs. 1. Molecular

frameworks", *Journal of Medicinal Chemistry*, Vol. 39, No. 15, pp. 2887–2893, jul 1996.

77. Van der Maaten, L. and G. Hinton, "Visualizing data using t-SNE.", *Journal of Machine Learning Research*, Vol. 9, No. 11, 2008.

78. DeLano, W. L. *et al.*, "PyMOL: An open-source molecular graphics tool", *CCP4 Newsletter on Protein Crystallography*, Vol. 40, No. 1, pp. 82–92, 2002.

79. Noort, A. R., K. P. van Zoest, E. M. Weijers, P. Koolwijk, C. X. Maracle, D. V. Novack, M. J. Siemerink, R. O. Schlingemann, P. P. Tak and S. W. Tas, "NF-$\kappa$B-inducing kinase is a key regulator of inflammation-induced and tumour-associated angiogenesis", *The Journal of Pathology*, Vol. 234, No. 3, pp. 375–385, 2014.

80. Ou, B., H. Sun, J. Zhao, Z. Xu, Y. Liu, H. Feng and Z. Peng, "Polo-like kinase 3 inhibits glucose metabolism in colorectal cancer by targeting HSP90/STAT3/HK2 signaling", *Journal of Experimental and Clinical Cancer Research*, Vol. 38, No. 1, pp. 1–12, oct 2019.

81. "Molecule.One", `https://molecule.one`.

82. Liu, C.-H., M. Korablyov, S. Jastrzebski, P. Włodarczyk-Pruszyński, Y. Bengio and M. H. Segler, "Retrognn: Approximating retrosynthesis by graph neural networks for de novo drug design", *arXiv preprint arXiv:2011.13042*, 2020.

83. Lin, Z., G. I. Winata, P. Xu, Z. Liu and P. Fung, "Variational transformers for diverse response generation", *arXiv preprint arXiv:2003.12738*, 2020.

84. Arroyo, D. M., J. Postels and F. Tombari, "Variational Transformer Networks for Layout Generation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13642–13652, 2021.

85. Kool, W., H. Van Hoof and M. Welling, "Stochastic beams and where to find them:

The gumbel-top-k trick for sampling sequences without replacement", *International Conference on Machine Learning*, pp. 3499–3508, PMLR, 2019.

86. Kool, W., H. van Hoof and M. Welling, "Ancestral Gumbel-Top-k Sampling for Sampling Without Replacement.", *Journal of Machine Learning Research*, Vol. 21, pp. 47–1, 2020.

87. Eikema, B. and W. Aziz, "Is map decoding all you need? the inadequacy of the mode in neural machine translation", *arXiv preprint arXiv:2005.10283*, 2020.

88. Horwood, J. and E. Noutahi, "Molecular design in synthetically accessible chemical space via deep reinforcement learning", *ACS Omega*, Vol. 5, No. 51, pp. 32984–32994, 2020.

89. Polishchuk, P., "Control of synthetic feasibility of compounds generated with CReM", *Journal of Chemical Information and Modeling*, Vol. 60, No. 12, pp. 6074–6080.

# APPENDIX A: CHEMICAL SPACE

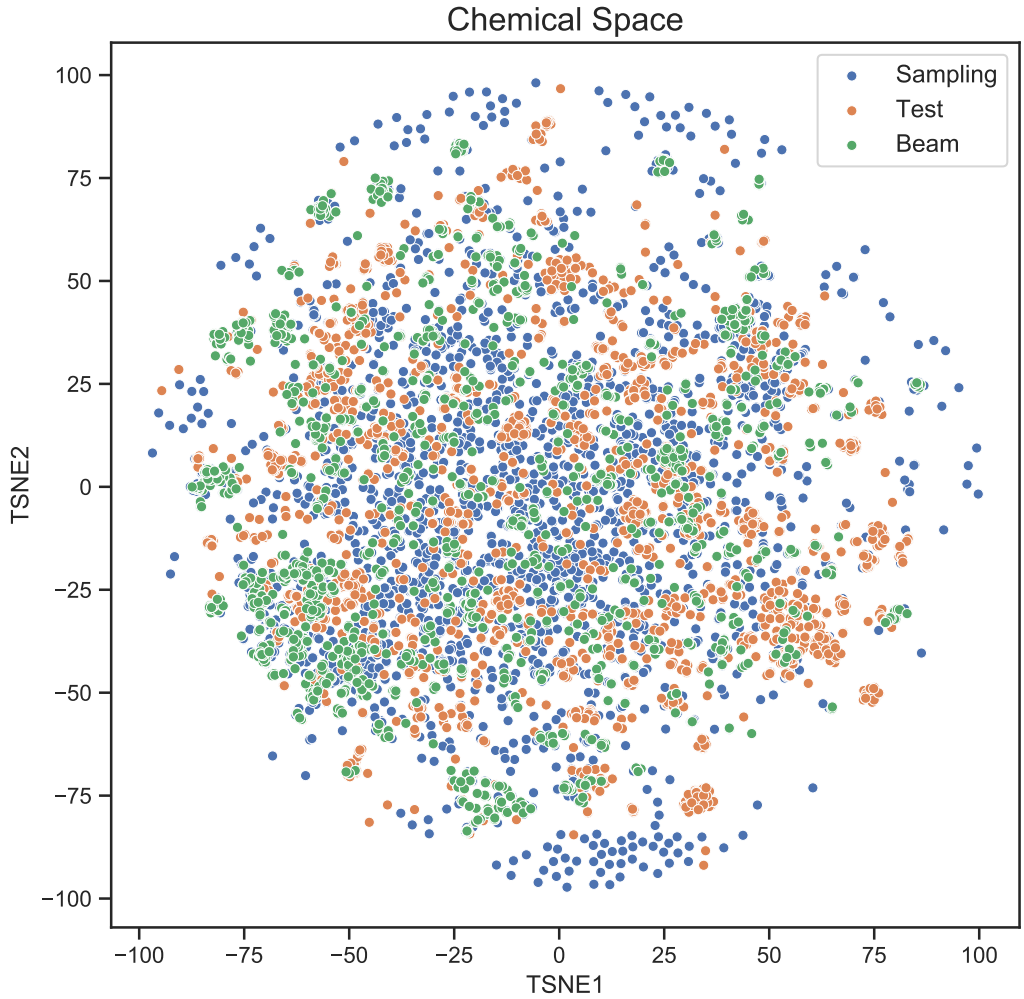


Figure A.1. t-SNE plot of generated compounds with EncDecLM model trained on all interactions and 5000 randomly sampled compounds from the test set.

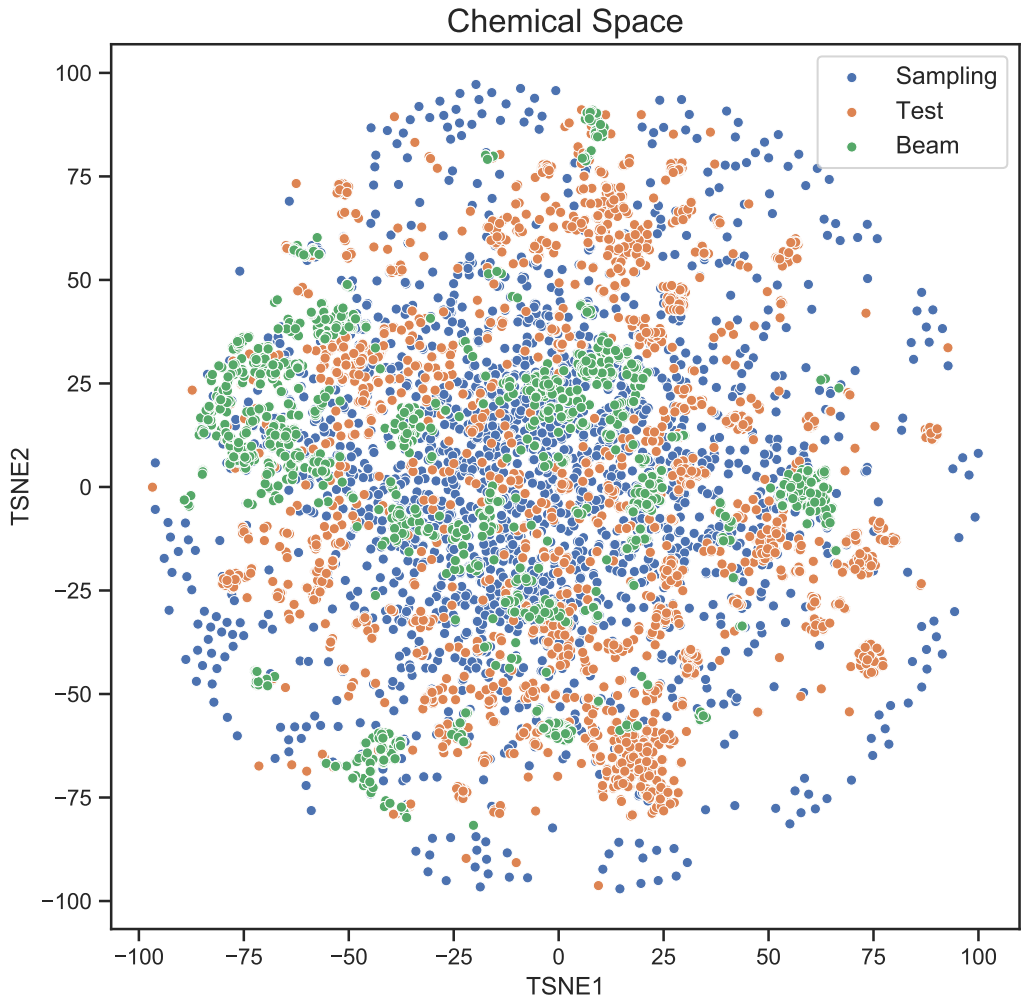


Figure A.2. t-SNE plot of generated compounds with T5 model trained on all interactions and 5000 randomly sampled compounds from the test set.

# APPENDIX B: DOCKING RESULTS

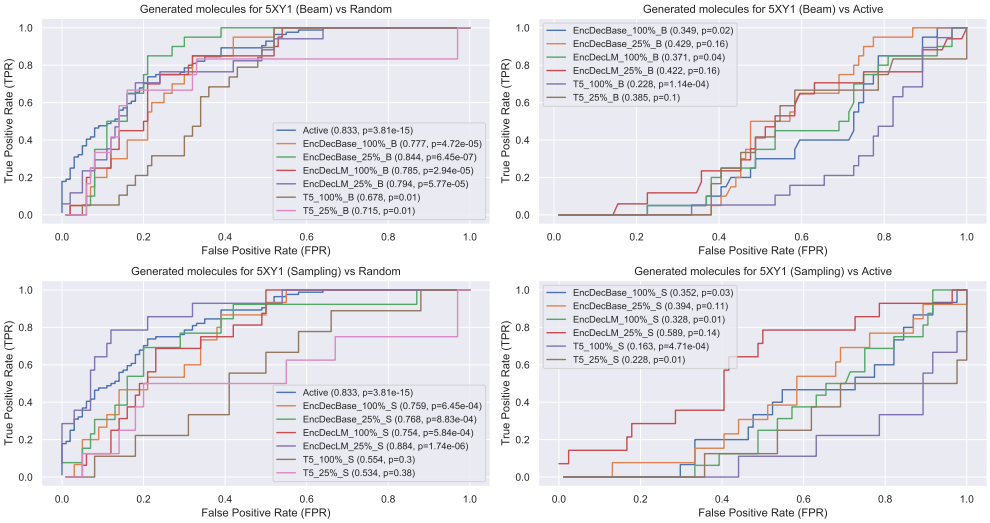


Figure B.1. ROC Curves for the comparison of the generated compounds for 5XY1 with random compounds and active compounds.

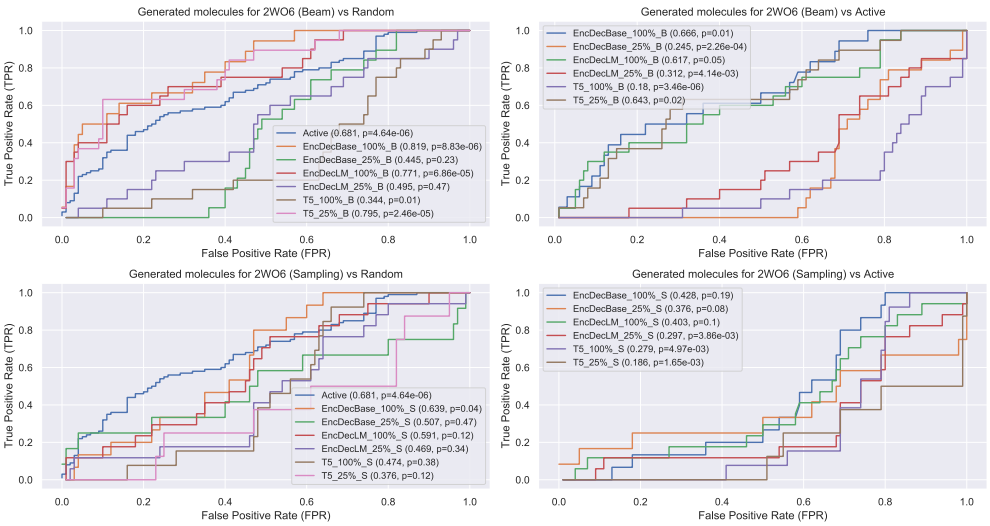


Figure B.2. ROC Curves for the comparison of the generated compounds for 2WO6 with random compounds and active compounds.

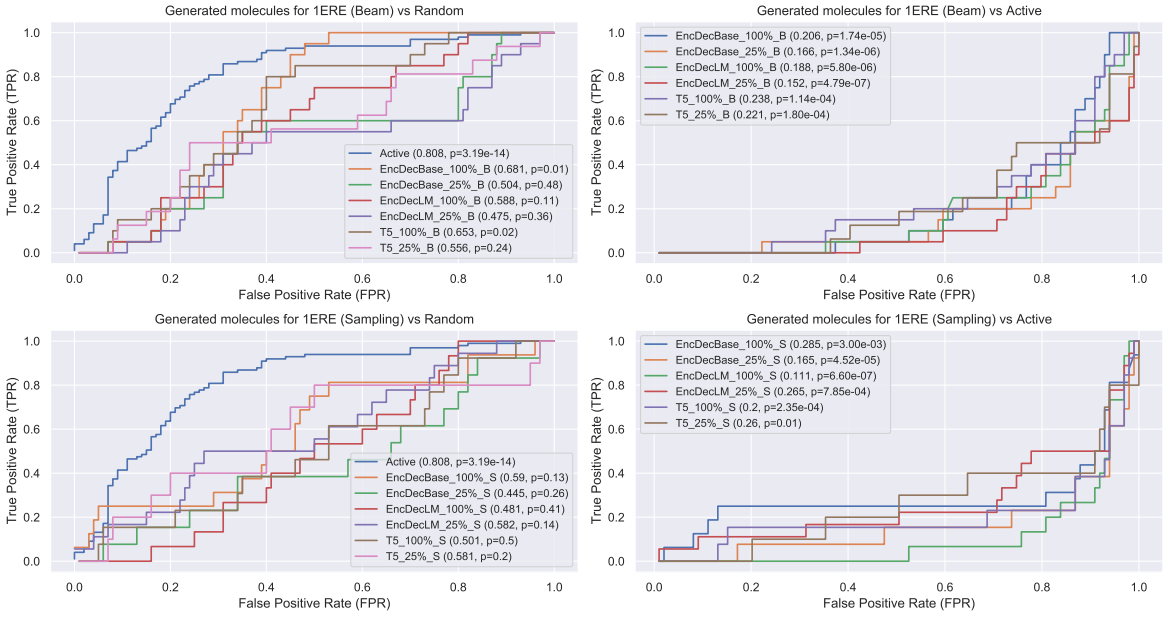


Figure B.3. ROC Curves for the comparison of the generated compounds for 1ERE with random compounds and active compounds.

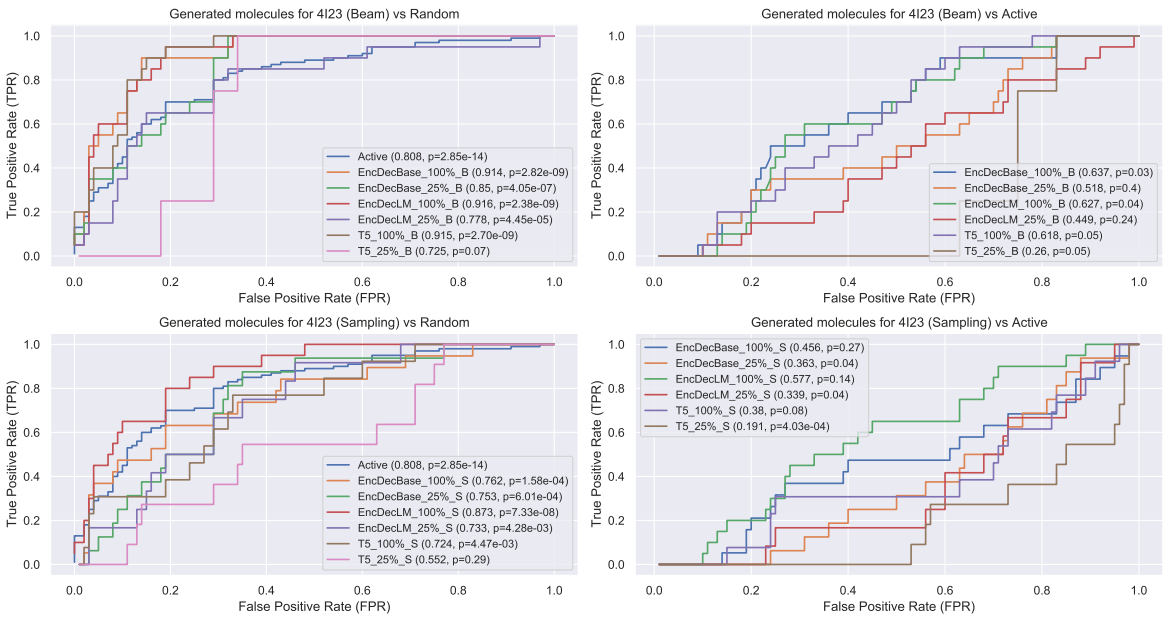


Figure B.4. ROC Curves for the comparison of the generated compounds for 4I23 with random compounds and active compounds.

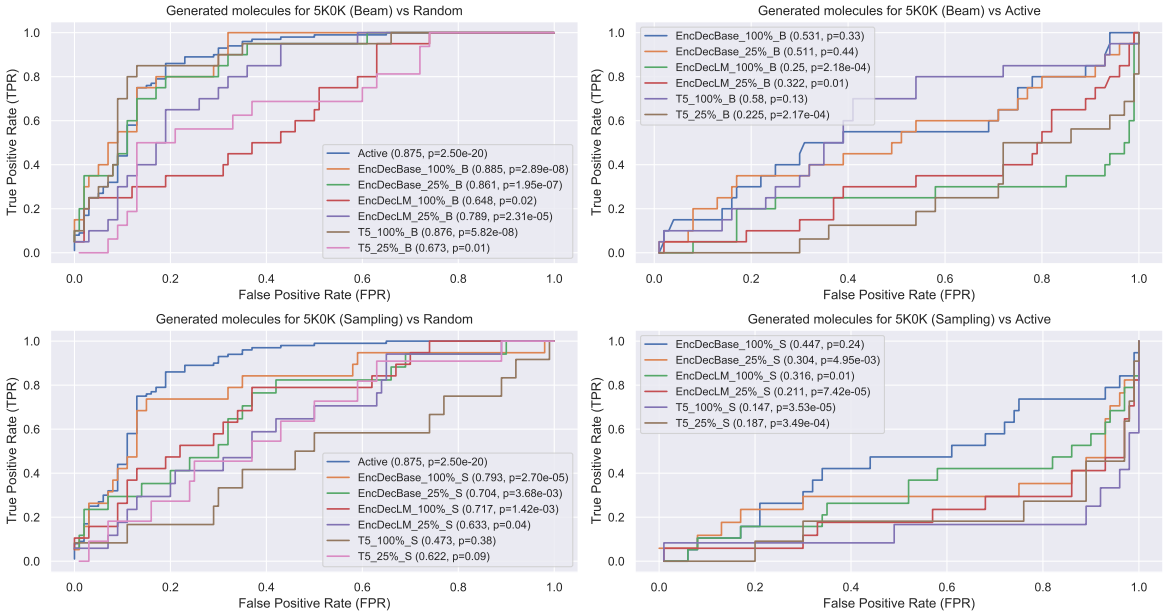


Figure B.5. ROC Curves for the comparison of the generated compounds for 5K0K with random compounds and active compounds.

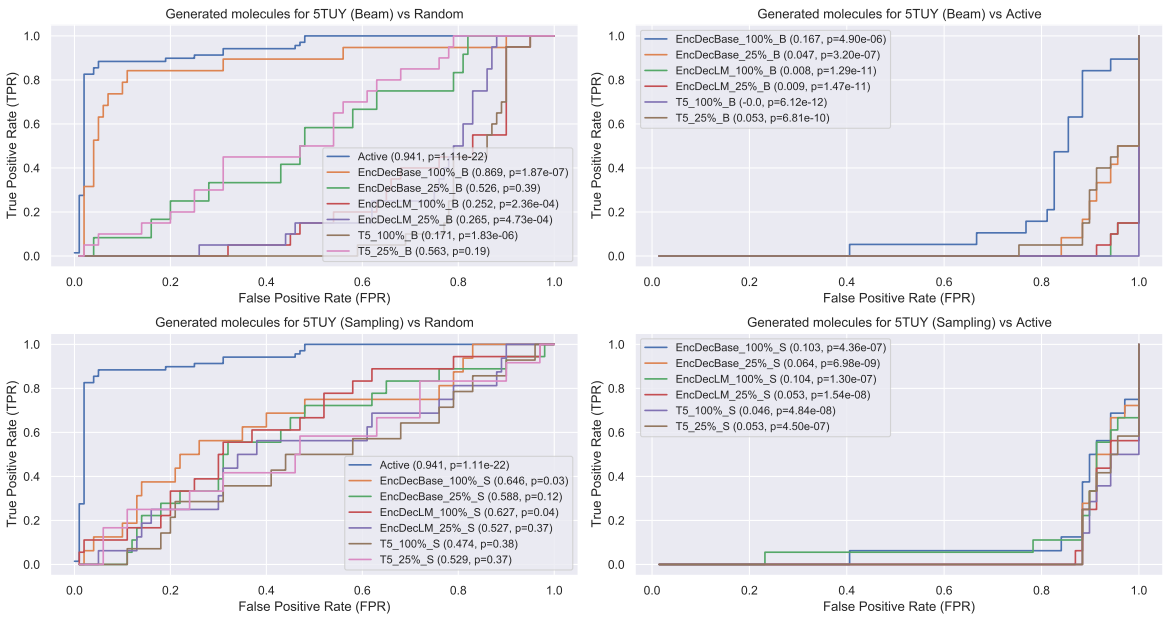


Figure B.6. ROC Curves for the comparison of the generated compounds for 5TUY with random compounds and active compounds.

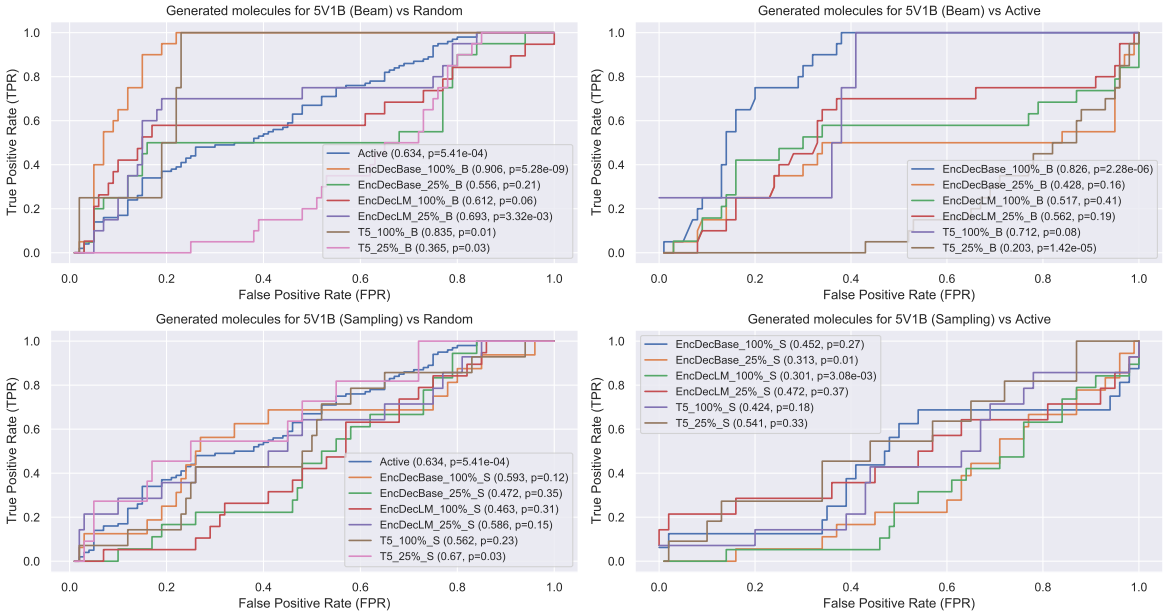


Figure B.7. ROC Curves for the comparison of the generated compounds for 5V1B with random compounds and active compounds.

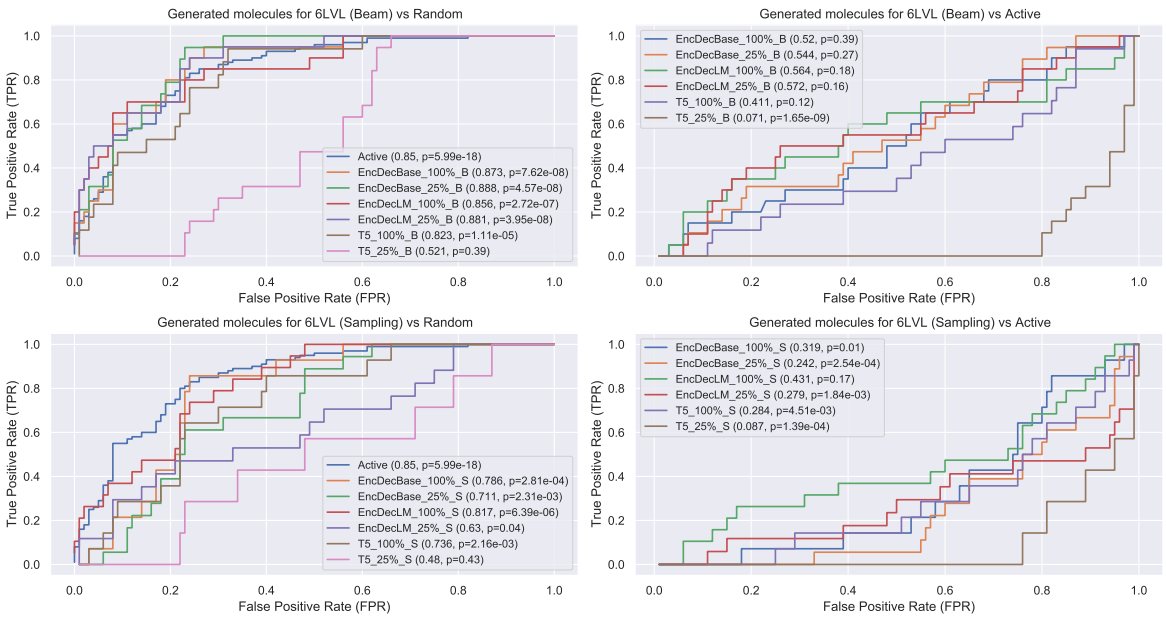


Figure B.8. ROC Curves for the comparison of the generated compounds for 6LVL with random compounds and active compounds.

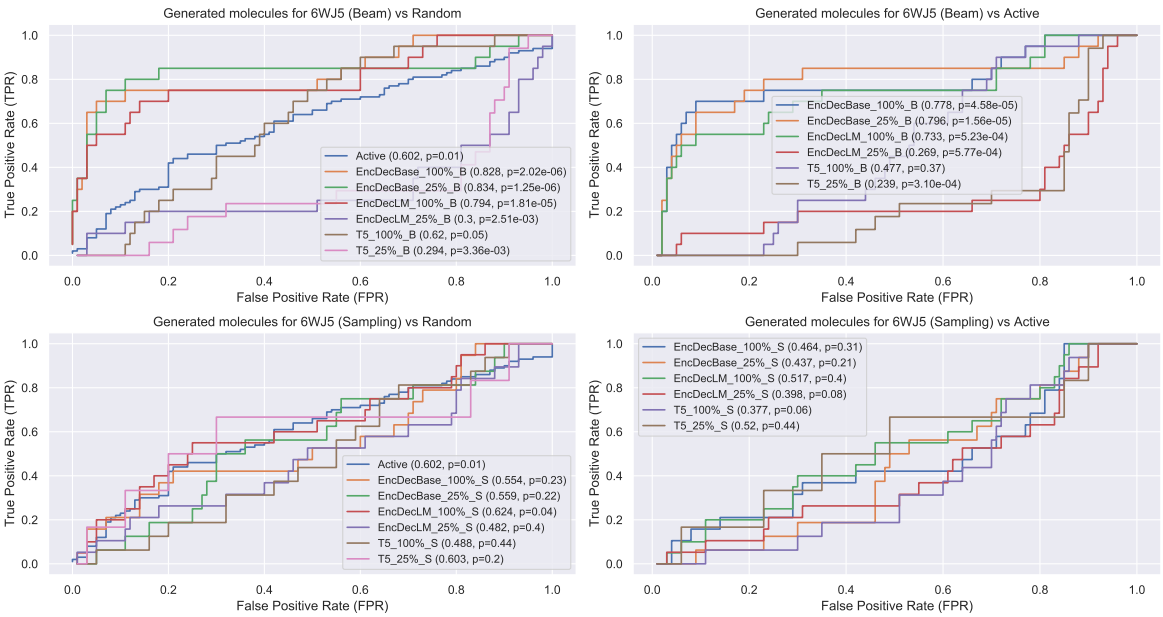


Figure B.9. ROC Curves for the comparison of the generated compounds for 6WJ5 with random compounds and active compounds.

# APPENDIX C: RETROSYNTHESIS PLANNING

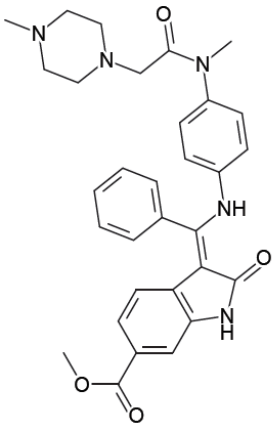


Figure C.1. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 2WO6.

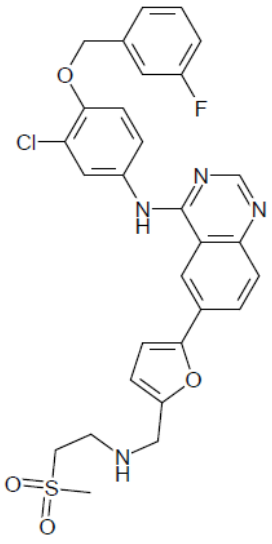


Figure C.2. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 4I23.

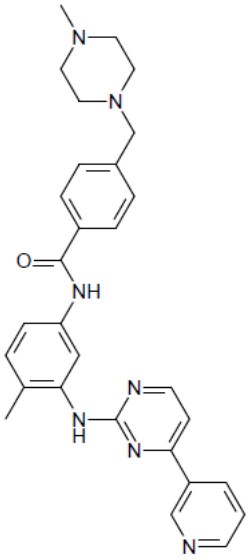

Figure C.3. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 5K0K.

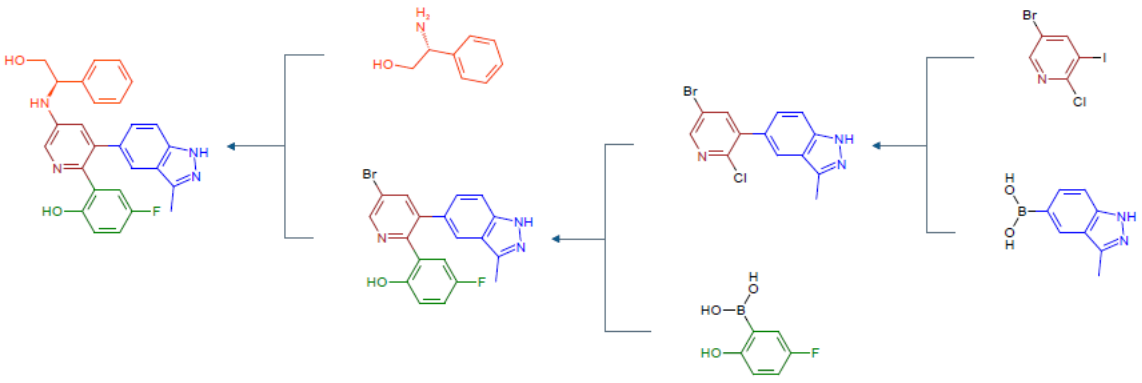

Figure C.4. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 6LVL.

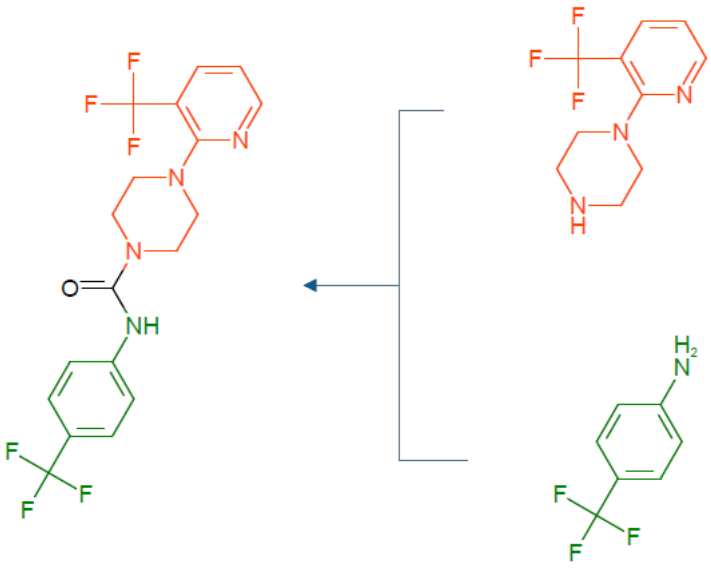


Figure C.5. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 6WJ5.

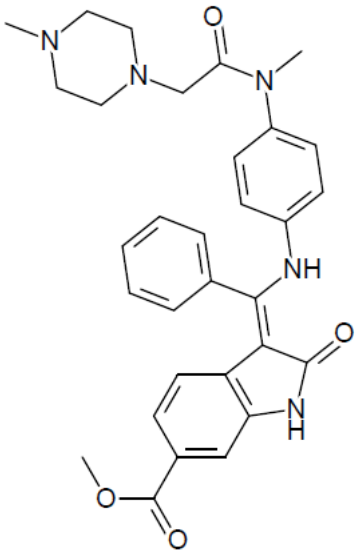


Figure C.6. Retrosynthesis planning of the generated compound with lowest synthesizability score for target 6Z1Q.