TARGET-BASED DRUG DISCOVERY THROUGH CONTRASTIVE LEARNING AND LATENT OPTIMIZATION

by

Burak Can Koban

B.S., Chemical Engineering, Boğaziçi University, 2021

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

2024

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Arzucan Özgür, for her great support and guidance. Whenever I felt there was no solution and I had reached a dead end, her positivity and encouragement reignited my belief in this journey.

I would like to thank my co-advisor, Elif Özkırımlı, for her critical insights and invaluable guidance throughout my thesis. Her contributions were essential in steering this work in the right direction.

A special thank you to my colleague and friend, Gökçe Uludoğan, for our countless discussions and her brilliant advices. I have learned so much from her and I am truly grateful for her support.

I would like to thank to my friend, Elvan Karasu, who understood me so well during this period. She experienced many of the same challenges and joys as I did, and her companionship was invaluable.

I gratefully acknowledge the support from TÜBİTAK ARDEB 1001 Program (Project No: 119E133). I also thank TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources) for providing computational resources. Additionally, I would like to thank Doğan Ulus for making the computational resource center available to his students.

I would like to thank my family for never stopping their belief in me. I dedicate this thesis to my family with all my love and gratitude.

Last but not least, my biggest thanks go to my greatest supporter, my life partner, Yağmur Bilge. She has always been there for me when I needed her, offering kindness, belief in me (sometimes even more than I had in myself), and love. All of this made

finishing this thesis possible. I'm grateful for her presence through the tough and joyful times, for sharing both sadness and laughter, for walking beside me quietly but understanding deeply, and for everything she brings into my life.

ABSTRACT

TARGET-BASED DRUG DISCOVERY THROUGH CONTRASTIVE LEARNING AND LATENT OPTIMIZATION

The development of novel compounds targeting proteins of interest is an important step in the process of drug discovery to cure diseases. Several studies have focused on generating target-specific molecules based on protein sequences alone. However, these models consider only interacting protein-ligand pairs, overlooking the informative value of non-interacting pairs. In this thesis, we introduce ConGen, a novel contrastive learning-based approach for targeted drug generation that effectively utilizes both interacting and non-interacting pairs. ConGen consists of two stages: contrastive learning and latent optimization. In the contrastive learning stage, ConGen learns to map protein and molecule representations into a shared space, clustering interacting molecules near their target proteins and non-interacting molecules farther away. In the latent optimization stage, ConGen uses this space to identify representations that are close to the representation of the target protein. This approach enables the generation of molecules that are specifically designed for a particular target by efficiently utilizing the arrangements in the shared latent space. Different contrastive learning loss functions are compared and the best performing functions are used to benchmark the model against other studies. The effectiveness of ConGen's contrastive learning and latent optimization stages are demonstrated using an ablation study. Moreover, the architecture of ConGen is easily adapted for the drug-target interaction prediction task, where it achieves results slightly better than the baseline. Lastly, a web service has been developed to deploy target-based drug generative models, enabling researchers to create and evaluate new drug molecules efficiently.

ÖZET

KARŞILAŞTIRMALI ÖĞRENME VE SAKLI VEKTÖR OPTİMİZASYONUNDAN YARARLANARAK HEDEF BAZLI İLAÇ KEŞFİ

Proteinleri hedef alan yeni bileşiklerin geliştirilmesi ilaç keşfi sürecinde önemli bir adımdır. Birçok çalışma, yalnızca protein dizilerine dayalı olarak hedefe özgü moleküller üretmeye odaklanmıştır. Ancak bu modeller yalnızca etkileştiği bilinen protein-ligand çiftlerini dikkate alır. Bunu yaparak etkileşmediği bilinen moleküllerden gelecek bilgiyi göz ardı eder. Bu tezde, hedefe yönelik ilaç üretimi için hem etkileştiği bilinen hem de etkileşmeyen protein-ligand çiftlerini kullanan ve karşılaştırmalı öğrenme temelli ConGen modelini sunuyoruz. ConGen'in iki aşaması vardır: karşılaştırmalı öğrenme ve saklı vektör optimizasyonu. Karşılaştırmalı öğrenme aşamasında ConGen, etkileştiği bilinen molekülleri hedef proteinlerinin yakınında ve etkileşime girmeyen molekülleri daha uzakta kümeleyerek, protein ve molekül temsillerini ortak bir uzayda göstermeyi öğrenir. Saklı vektör optimizasyonu aşamasında ConGen bu uzayı hedef proteinin temsiline yakın temsilleri bulmak için kullanır. Bu yaklaşım, moleküllerin ve proteinlerin paylaştığı uzayı verimli bir şekilde kullanarak, belirli bir hedef için tasarlanmış moleküllerin üretilmesini sağlar. En iyi öğrenme hedefini belirlemek için farklı karşılaştırmalı öğrenme fonksiyonları karşılaştırılmıştır. En iyi performans gösteren öğrenme fonksiyonları modeli diğer çalışmalarla karşılaştırmak için kullanılmıştır. ConGen'in karşılaştırmalı öğrenme ve gizli optimizasyon aşamalarının etkinliği, bu iki bileşenin kontrollü olarak çıkarıldığı deneyler ile gösterilmiştir. Dahası ConGen'in mimarisi ilaç-hedef etkileşimi tahmini için kullanılmış ve sonuçlar bir miktar iyileştirilmiştir. Son olarak hedefe dayalı ilaç üretim modellerinin çevrimiçi bir kullanılabilecek bir web servisi geliştirilmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT v
ÖZET vi
LIST OF FIGURES ix
LIST OF TABLES xi
LIST OF SYMBOLS xii
LIST OF ACRONYMS/ABBREVIATIONS xiii
1. INTRODUCTION 1
2. BACKGROUND 6
2.1. Drug Discovery 6
2.2. Proteins 7
2.3. Ligands 7
2.4. Projection 8
2.5. Contrastive Learning 9
2.5.1. Losses 9
2.5.1.1. Contrastive Loss 10
2.5.1.2. Triplet Loss 10
2.5.1.3. N-pair Loss 10
2.5.1.4. InfoNCE Loss 10
2.6. Autoencoders 11
2.7. Graph Neural Networks 11
2.8. Transformers 12
3. MATERIALS AND METHODS 13
3.1. Data 13
3.2. Model 15
3.2.1. Classification Task 16
3.2.2. Contrastive Learning 17
3.2.3. Latent Optimization 18

3.2.4. Molecule Generation 19
3.3. Implementation Details 20
3.4. Evaluation 20
3.4.1. Precision and Recall 20
3.4.2. Metrics 21
3.4.3. t-Distributed Stochastic Neighbor Embedding Visualisation . . . 22
4. RESULTS 23
4.1. ConGen 23
4.1.1. DTI Prediction Task. 23
4.1.2. Loss Function Comparison. 25
4.1.3. Ablation Study 26
4.1.4. Baseline Comparison 27
4.1.5. Space Visualization 28
4.2. Web Service 30
5. DISCUSSION 33
6. CONCLUSION 35
REFERENCES 36
APPENDIX A: DECODING STRATEGY 44
APPENDIX B: CONTRASTIVE LEARNING T-SNE RESULTS 45
APPENDIX C: LATENT OPTIMIZATION T-SNE RESULTS 47

LIST OF FIGURES

Figure 2.1. Summary of the drug discovery process. 6

Figure 3.1. Boxplot analysis of the data before and after removing outliers. . . 14

Figure 3.2. Model pipeline of ConGen. 16

Figure 3.3. Illustration of contrastive learning. 18

Figure 3.4. Illustration of latent optimization. 19

Figure 4.1. Average cosine similarities between interacting molecules and proteins. 24

Figure 4.2. Average cosine similarities between non-interacting molecules and proteins. 24

Figure 4.3. A comparative analysis of ConGen model with different losses in terms of FCD per target distribution. Lower FCD is better. 25

Figure 4.4. A comparative analysis of ConGen model with baseline models in terms of FCD per target distribution. Lower FCD is better. 29

Figure 4.5. Visualization of the shared space before and after contrastive learning for protein with UniProt ID Q8CA95. 30

Figure 4.6. Visualization of latent space adjustment for protein UniProt ID P0A725. 31

Figure 4.7. Web service. 32

Figure A.1. Neighborhood of generated molecules. 44

Figure B.1. Protein with UniProtID P97629. 45

Figure B.2. Protein with UniProtID Q6GG09. 45

Figure B.3. Protein with UniProtID Q9Y6K0. 46

Figure C.1. Protein with UniProtID P09769. 47

Figure C.2. Protein with UniProtID P36896. 48

Figure C.3. Protein with UniProtID P49286. 49

Figure C.4. Protein with UniProtID P05981. 50

Figure C.5. Protein with P13612. 51

Figure C.6. Protein with Q16584. 52

LIST OF TABLES

Table 3.1. Training, validation and test data summary. 15

Table 4.1. AUROC and AUPR results for different encoding methods. 23

Table 4.2. Comparison of different loss functions on model's performance. . . 25

Table 4.3. Ablation study results. 27

Table 4.4. Comparison between ConGen and baseline models for MOSES metrics. 28

Table 4.5. Comparison between ConGen and baseline models for molecular property distance metrics. 28

LIST OF SYMBOLS

$\mathbf{a}$	Anchor point
$\mathbf{b}$	Bias
c	Density ratio
d_k	Scaling factor
f	Encoding function
$\mathbf{h}$	Hidden layer
IC_{50}	Half maximal inhibitory concentration
$\mathbf{K}$	Keys
l	Latent
$\mathcal{L}_{\mathrm{i}}$	Loss
$\mathbf{n}$	Negative point
p	Protein
$\mathbf{p}$	Positive point
$\mathbf{Q}$	Query
$\mathrm{proj}_S(\mathbf{v})$	Projection of v onto space S
$\mathbf{u}_i$	Basis vector
$\mathbf{v}$	Vector
$\mathbf{V}$	Values
$\mathbf{W}$	Weight matrix
$\hat{y}_i$	Prediction label
$\mathbf{x}$	Input samples
y	Labels
y_i	Actual label
ϵ	Margin

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
AUPR	Precision–recall curve
AUROC	Area under the receiver operating characteristic
BART	Bidirectional and Auto-Regressive Transformer
BRICS	Breaks into Chemical Components
DNA	Deoxyribonucleic acid
DTI	Drug Target Interaction
FCD	Frechet ChemNet Distance
FN	False Negatives
FP	False Positives
FBDD	Fragment-based Drug Design
GNN	Graph Neural Network
IQR	Inter-quartile ranges
IntDiv	Internal Diversity
logP	Logarithm of the partition coefficient
MCF	MCF Filters (MOSES)
MCTS	Monte Carlo tree search
NP	Natural Product-likeness
PAINS	Pan Assay Interference Compounds
Q1	First Quartile
Q3	Third Quartile
QED	Quantitative Estimate of Drug-likeness
ROC	Receiver Operating Characteristic
SA	Synthetic Accessibility
Scaf	Scaffold similarity
SMILES	Simplified Molecular Input Line Entry System

SNE	Stochastic Neighbor Embedding
SNN	Single Nearest Neighbour
t-SNE	t-Distributed Stochastic Neighbor Embedding
TP	True Positives
TN	True Negatives
Uniq	Uniqueness
UniProtID	Universal Protein Resource Identifier

1. INTRODUCTION

Discovering a drug-like molecule with the desired properties is a difficult and laborious process. The size of the drug-like molecule space is immense, approximately 10^{60} [1], which significantly increases the challenge of discovering new molecules. Furthermore, developing a drug fully can take 12 years or more [2]. This is a long process when it comes to deal with the rapidly evolving nature of pathogenic substances. The primary goal of the initial phases of drug discovery is to identify a lead molecule that exhibits binding affinity to a specific target protein and possesses the desired pharmacological and chemical properties [3]. This task and the associated challenges can be overcome by developing new computational methods.

Computational methods have shown remarkable success in drug discovery [4]. Especially, deep learning methods in target-specific drug discovery have seen a notable increase. LIMO [5], for example, leverages a variational auto-encoder to generate a latent space, combined with a property prediction network, to accelerate the targeted drug discovery process. This approach involves reverse optimization of the generated latent space which enables the decoding of molecules with desirable properties. Another work [6] uses graph attention model to learn active site of proteins to use them in conditional molecule generation and employs a bioactivity prediction model is for optimization. CLigOpt model employs fragment-based drug design (FBDD) to navigate the vast molecular space efficiently and generate molecules with desired properties from given fragment pairs by utilizing graph-based deep generative model [7]. Additionally, FREED++ [8], an advancement of [9], aims to utilize reinforcement learning to generate molecules by providing docking score as a reward.

Besides molecule generation, recent advancements in drug-target interaction (DTI) prediction models play a crucial role in drug discovery. An early work, DeepDTI [10], utilizes deep neural networks for drug-target interaction prediction. DeepDTA [11], on the other hand, employs convolutional neural networks to predict drug-target interac-

tions. MolTrans [12], for example, uses transformers to gather contextual information from proteins and molecules and also leverages both labeled and unlabeled data. HyperPCM encodes drug compounds and target proteins alongside protein context to enhance protein embedding and achieves state-of-the-art performance [13].

In drug discovery, encoding molecule representation is a crucial step. Transformers are extensively used in this field to learn these representations. For instance, MegaMolBART [14], based on the Bidirectional and Auto-Regressive Transformer (BART) architecture, is specifically trained for encoding molecules into embeddings and for decoding embeddings. The encoder and decoder components of the model can be utilized separately in predictive models. Another significant study, ChemBERTa [15], utilizes Bidirectional Encoder Representations from Transformers and is trained on 77 million unique Simplified Molecular Input Line Entry System (SMILES) sequences, making it suitable for any task that requires molecule representation. Additionally, MolT5 [16] focuses on translation task between molecule descriptions and SMILES. This approach enables the molecule encoder and decoder to be used for extracting and refining molecular representations.

Recently, the generation of target specific molecules has been a focus of many studies. A significant area of research involves structure-based drug design methods that utilize protein structure. These methods typically use autoregressive models [17–19] or diffusion models [20–22]. However, they are limited by the scarcity of structures involving ligands in bound form and are known to generate physically implausible molecules, such as those geometrically clashing with the target protein [21,22]. Another line of research leverages active compounds or docking experiments to direct molecule generation. These studies are generally trained with huge dataset of molecules and then fine-tuned for target-based molecule generation. However, these studies face challenges: they depend on the availability of molecules active against specific targets (often lacking for new proteins), rely on predictive models that may result in the generation of molecules that, despite having high scores, contain synthetically infeasible, or highly uncommon substructures [23], or require computationally expensive dock-

ing experiments. More recently, several studies have developed models that require only the protein sequence for target generation. [24] approached targeted drug design as a "translation from protein language to chemical language", using a transformer model trained from scratch on interacting protein-ligand pairs. Adopting a similar approach, [25] exploited pretrained protein and chemical language models trained on abundance of sequences and proposed two warm start strategies for fine-tuning these models on interacting protein-ligand pairs. They presented the EncDecBase model, employing a one-stage strategy where the initialized model is directly fine-tuned on protein-ligand pairs [25]. Additionally, they developed the EncDecLM model, which use a two-stage strategy, beginning with an initial fine-tuning using compounds from MOSES [26]. However, these studies only utilize data on interacting protein-ligand pairs, overlooking valuable information from non-interacting pairs, which we believe could enhance target-specific generation. Conversely, [27] adopted a similar approach by modifying the transformer model and coupling it with a Monte Carlo tree search (MCTS), guided by docking experiments. While innovative, this approach is computationally intensive.

In recent years, contrastive learning has made significant strides across various fields, including natural language processing and computer vision [28]. This method aims to organize a representational space where similar items are grouped together, while dissimilar ones are spaced apart. This technique has also been embraced in drug discovery [29, 30]. For example, ConPLex [29] introduced a contrastive learning-based method where proteins and drugs are mapped to a shared latent space. In this approach, the protein is considered the anchor point, while interacting molecules are treated as positive pairs. A very similar molecule to the positive molecule, called a decoy, is considered a negative sample of a triplet. This method effectively differentiates between actual interacting pairs and decoys, resulting in high specificity and generalization. Our methodology builds upon this by using contrastive learning to structure a shared space for protein-ligand pairs, viewing the proximity of proteins and ligands as an affinity indicator. MoleculeSTM [30], in contrast, focuses on aligning molecule structures with their corresponding textual descriptions through contrastive

pretraining. This model is then employed for various applications, including text-based molecule editing and molecular property prediction. In the molecule editing task, latent optimization is used to align the embedding of a textual description with that of a chemical structure, resulting in the generation of a molecule that meets the described requirements. Similarly, our research applies latent optimization to align randomly generated latent projections more closely with the target protein's projection. This approach aims to facilitate the generation of molecules that are precisely tailored to specific target proteins by leveraging this optimized latent space.

In this thesis, a model architecture is developed that combines protein and molecule encoders with projectors to map their representations into a shared space. This shared space is utilized for both DTI prediction and targeted drug generation. For DTI prediction, the similarity between drug and protein projections is used as an affinity score. To establish the shared space of DTI prediction, a binary classification task is employed, where true labels are derived from actual data, and predicted labels are generated based on the predicted affinity scores.

The model architecture was further developed into ConGen, a novel targeted molecule design model. To the best of our knowledge, ConGen uniquely integrates pretrained protein and molecule models with contrastive learning and latent optimization techniques, enhancing its ability to accurately predict and design target-specific drug molecules. ConGen leverages contrastive learning to create a shared space for drug molecules and proteins. This process structures the shared space so that interacting protein-ligand pairs are positioned closely, while non-interacting pairs are kept distant. Subsequently, latent optimization is employed to refine the latent representation, starting from a random state, to align more closely with that of the target protein. We hypothesize that molecules emerging from this optimized latent space will exhibit an affinity for the target protein. Notably, our technique relies solely on the protein's amino acid sequence, eliminating the need for protein structure or target-specific molecules. Furthermore, it is relatively lightweight, leveraging pretrained models and introducing a significantly smaller number of trainable parameters.

We employ a graph neural network (GNN) based model, MoLeR [31], for encoding and decoding molecules, and a transformer-based model, ProtBERT [32], for protein encoding, leveraging their strengths of being pretrained on a large dataset. The model was optimized using binary cross entropy loss for the DTI prediction task. We compared the effects of three different molecular encoding methods—ChemBERTa, MegaMolBART, and MolT5—with a baseline, ConPLex, to evaluate their impact on DTI.

The model is trained using three different contrastive learning loss functions for targeted molecule design task: triplet loss, N-pair loss and InfoNCE loss . We compared the performances of these loss functions. Moreover, we compare our model, ConGen, against EncDecLM using previously unseen proteins, demonstrating that ConGen exhibits competitive performance. We conducted an ablation study to highlight the effectiveness of ConGen's individual stages: contrastive learning and latent optimization. Lastly, we created a web service which can be used to deploy target specific drug generation models.

2. BACKGROUND

2.1. Drug Discovery

Drug discovery is an expensive and time consuming task. A development of pharmaceutical medicine can take anywhere from 2 to 12 years from the discovery of the drug to its market release. The process of drug discovery can be divided into 5 different stages [33]: drug discovery, preclinical research, clinical trials, reviewing, post-market safety monitoring. Determining the target protein and identifying potential drug molecules is the initial step of the process which takes approximately 6 to 7 years. As a result of this stage, the lead molecules are narrowed down from the broader chemical molecule space to the more specific possible drug molecule space. This step follows with the preclinical research where the experiments are conducted in vitro (inside experimental tubes) and in vivo (inside animal subjects). The next step is the clinical research which is carried out in four phases. In clinical research, the number of candidates with the disease condition gradually increases with each phase. Phase 1 typically involves 20 to 100 candidates, whereas Phase 4 includes several thousand volunteers. When a new drug application is received, it is reviewed by an independent intuition. Then, depending on the outcomes of the previous stage, the drug is either released to the market or not. If it is decided for market release, it is monitored during the post-market period to ensure safety. The process of drug discovery is summarized on Figure 2.1 [33].



Figure 2.1. Summary of the drug discovery process.

2.2. Proteins

Proteins are macro-molecules composed of 20 different amino acids. These building units of proteins, known as amino acids, are linked to each other with peptide bonds and the sequence of the amino acids are determined by the genes. Proteins are used various functions in the living organisms across facilitate metabolic reactions, forming the cells and tissues and transporting molecules across the cells [34]. In living organisms, deoxyribonucleic acid (DNA) is composed of genes. Genes determine the sequence of amino acids in proteins. These sequences can influence the three-dimensional (3D) shape of proteins, which in turn dictates their functionality [35]. While the 3D structures of proteins are available, their one-dimensional (1D) amino acid sequences are also accessible. Proteins can be represented as 1D strings where each amino acid is denoted by a different letter.

2.3. Ligands

Ligands are molecules that have a high binding affinity for specific sites on proteins, inhibiting biological processes through these interactions. For therapeutic applications, binding is essential for identifying potential drug molecules for target proteins. Ligand molecules can be represented in one, two, or three-dimensional (1D, 2D, 3D) formats.

The most common method to describe molecules is through well-known molecular formulas. However, molecules can also be represented in 2D, which is commonly known as molecular graphs. In graph representations, the vertices represent different atomic molecules and the edges correspond to the types of bonds between them. The graph representation can be converted into the Simplified SMILES, which provides a textual representation of the molecule.However, there can be multiple SMILES representations for a single graph representation of a molecule. This can be both an advantage and a disadvantage, depending on the task at hand. This variety can be used as augmented data but some studies might view this as an ambiguity. To acquire a unique SMILES

for each graph representation, the canonicalization process can be utilized to achieve canonical SMILES. The most realistic way to represent molecules is undoubtedly the 3D method where the representation includes information about the locations of atoms in three-dimensional space. However, complexity and computational demands increase from 1D to 3D. Therefore, utilizing lighter and simpler representations like 1D more practical in certain scenarios.

2.4. Projection

In mathematics projection is defined as mapping a vector from one space to another through linear transformation. Given a vector $\mathbf{v}$ in a space $\mathbb{R}^n$ and a subspace S defined by a basis of vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$, the projection of $\mathbf{v}$ onto S is given in Equation 2.1. In this equation, $\mathbf{v} \cdot \mathbf{u}_i$ is the dot product of $\mathbf{v}$ and $\mathbf{u}_i$, and $\mathbf{u}_i \cdot \mathbf{u}_i$ is the dot product of $\mathbf{u}_i$ with itself, providing the normalization factor. The projection function is described as

$$\text{proj}_S(\mathbf{v}) = \sum_{i=1}^{k} \frac{\mathbf{v} \cdot \mathbf{u}_i}{\mathbf{u}_i \cdot \mathbf{u}_i} \mathbf{u}_i. \tag{2.1}$$

Space projection is a crucial concept in machine learning. Space projection often refers to the transformation of data from a high-dimensional space to a lower-dimensional space. This reduced space often called latent space or feature space. The aim of this transformation is capturing important information while ignoring irrelevant information in the latent space. It commonly used for simplifying complex data, enhancing visualization, and improving the performance of machine learning models.

A neural network can learn to project data. A feedforward neural network progressively transforms input data through combination of linear and non-linear operations. For a neural network with input $\mathbf{x} \in \mathbb{R}^n$, we want to project this input onto a k-dimensional space where $k < n$. The projection is described in Equation 2.2 where $\mathbf{W}$ is a weight matrix of size $k \times n$, $\mathbf{b}$ is a bias vector of size k and f is a non-linear activation function. This layer $\mathbf{h}$ represents the projected data in the k-dimensional

latent space. The goal of training the network is to adjust $\mathbf{W}$ and $\mathbf{b}$ such that $\mathbf{h}$ captures the most important features of $\mathbf{x}$ for the task at hand. The projected layer is defined as

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b}). \tag{2.2}$$

2.5. Contrastive Learning

The aim of contrastive learning is to create a shared space where similar samples are positioned close together and dissimilar samples are kept far apart. For example, imagine a shared space formed by image embeddings. In this space, images of cats should be clustered together, as should images of dogs, but the embeddings for cats and dogs should be distinctly separated from each other. This analogy can be expanded to other domains as well.

Contrastive learning is possible in both supervised and unsupervised settings. Supervised contrastive learning leverages labeled data to train models. The labels guide the model in determining whether the data points should be paired positively or negatively. On the other hand, in unsupervised scenarios, contrastive learning serves as an effective tool for self-supervised learning. In this approach, positive and negative pairs are derived from the unlabeled data. For example, data augmentation is a commonly used technique in contrastive learning to create these pairs. Augmented versions of a sample are treated as positive pairs, whereas different samples are considered negative pairs. However, this approach has some vulnerabilities, such as the risk of including false negatives among the negative data.

2.5.1. Losses

There are many objective functions in contrastive learning, few of them listed here. Early approaches to contrastive learning considers one positive and one negative sample for calculating loss. However, later approaches includes multiple positives and negatives in the loss calculation process.

2.5.1.1. Contrastive Loss. Contrastive loss [36] is one of the earliest and fundamental loss in the contrastive learning field. In contrastive loss, the aim is maximizing the similarity between same labeled pairs while minimizing similarity for different labeled samples. Given x represents the input samples, y denotes the labels, and ϵ is a hyperparameter that regulates the distance between different classes, the contrastive loss described as

$$\mathcal{L}_{\text{contrast}}(\mathbf{x}_i, \mathbf{x}_j, \theta) = \begin{cases} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 & \text{if } y_i = y_j \\ \max(0, \epsilon - \|\mathbf{x}_i - \mathbf{x}_j\|_2)^2 & \text{if } y_i \neq y_j \end{cases}. \tag{2.3}$$

2.5.1.2. Triplet Loss. For triplet loss [37] calculation, the data need to be structured into triplets consisting of an anchor point $\mathbf{a}$, a positive point $\mathbf{p}$, and a negative point $\mathbf{n}$. Positive samples and anchor belongs to same class while negative samples belongs to different class. Given the margin parameter ϵ is set to ensure a minimum difference between the distances of similar and dissimilar pairs, the triplet loss function described as

$$L_{triplet}(\mathbf{a}, \mathbf{p}, \mathbf{n}) = \frac{1}{N} \sum_{i=1}^{N} \max(d(\mathbf{a}_i, \mathbf{p}_i) - d(\mathbf{a}_i, \mathbf{n}_i) + m, 0). \tag{2.4}$$

2.5.1.3. N-pair Loss. N-pair loss [38] extends the concept of triplet loss by considering multiple positive and negative samples for each anchor. Instead of concentrating on a single data point, N-pair loss aims to maximize the distance between an anchor and multiple negative samples while also minimizing the distance between the anchor and multiple positive pairs. N-pair loss function is described as

$$\mathcal{L}_{\text{N-pair}}(\mathbf{a}, \mathbf{p}, \{\mathbf{n}_i\}_{i=1}^{N-1}) = \log\big(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{a})^\top f(\mathbf{n}_i) - f(\mathbf{a})^\top f(\mathbf{p}))\big). \tag{2.5}$$

2.5.1.4. InfoNCE Loss. Information noise contrastive estimation (InfoNCE) [39] loss is originally inspired from noise contrastive estimation [40]. The InfoNCE is based on viewing contrastive learning as a binary classification problem. For that purpose,

it utilizes categorical cross-entropy loss to distinguish the positive sample from the unrelated noise samples. Following equation is describing the InfoNCE loss function. Given f and c stands for density ratio and context. the loss function optimizes negative log probability, described as

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}\Big[\log \frac{f(\mathbf{x}, \mathbf{c})}{\sum_{\mathbf{x}' \in X} f(\mathbf{x}', \mathbf{c})}\Big]. \tag{2.6}$$

2.6. Autoencoders

Autoencoders [41] are special type of neural network used for unsupervised learning. The structure is composed of two units: the encoder and the decoder. The middle layer is the bottleneck layer which is also known as latent space or hidden layer with the smallest dimension. The encoder maps the input to a latent space representation and learns to compress input data into a smaller and denser representation. Next, the decoder learns to reconstruct the input data from the encoded representation. Autoencoders are trained to minimize a loss function that measures the difference between the input and its reconstruction. Given n is the number of samples, $\mathbf{x}_i$ is the actual input, and $\hat{\mathbf{x}}_i$ is the reconstructed input, a common choice for the loss function is the mean squared error (MSE), defined as

$$\text{Loss} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2. \tag{2.7}$$

2.7. Graph Neural Networks

Graphs are important data structure that can be leveraged in various areas. Images, texts, social networks or molecules can be represented as graphs. A graph is composed of nodes (vertices), which can represent entities such as people in a social network or atoms in molecules, and edges, which denote the relationships between these nodes, like friendships in social networks or chemical bonds in molecular structures. As another example, in images and texts data pixels and words serve as nodes with edges

connecting adjacent pixels and consecutive words, respectively. Those are strong representations for solving complicated problems like drug discovery, image segmentation or fraud detection.

In machine learning, neural networks have achieved significant success in processing structured data. However, for handling unstructured data, more specialized form of neural network is required, known as graph neural networks (GNNs) [42]. It is a special type of neural network that operates on graphs. GNNs operate by employing a mechanism known as message passing which is essential for aggregating and transforming information across the nodes of the graph. In message passing process, each node passes information to and from its adjacent nodes. This message passing continues for few rounds aggregating a final state while the all the message passing information is used to define final state of each node embedding. By leveraging graph convolutional layers and the states of the graph, an embedding space of the data can be learned.

2.8. Transformers

Transformers [43] employ an encoder-decoder architecture where the encoder maps input data to continuous representation and decoder generates outputs from this space. Both the encoder and decoder are formed by several identical layers that enhance the model's ability to process data. Attention mechanism is an core idea of these layers which use queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$). This mechanism helps to focus on important parts of the input data which results in better generation of outputs. These 3 types of vectors are utilized to determine the relevance of different parts of the input data. Given d_k is a scaling factor that helps normalization [43], the attention function can be defined as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2.8}$$

3. MATERIALS AND METHODS

3.1. Data

We utilized BindingDB [44], specifically BindingDB_202403 version, containing measured binding affinities between protein-ligand pairs to construct our dataset. To create a reliable dataset, we filtered this database as follows. We excluded protein-ligand interactions that lack essential information for this study, including target protein sequence, ligand SMILES, and UniProt database identifier. Additionally, we omitted interactions involving multi-chain proteins for simplicity. We canonicalized SMILES strings to ensure each molecule is represented by a unique SMILES notation. This is important because we use SMILES strings to categorize protein-ligand interactions as interacting or non-interacting. Next, we used the IC_{50} affinity score to label interactions between proteins and ligand molecules. Consequently, interactions without IC_{50} values were eliminated. Interactions with an IC_{50} higher than 100 nM were classified as interacting, whereas those with lower values were considered non-interacting [45].

The filtered dataset remains imbalanced in terms of protein representation. While some proteins are associated with thousands of interactions, others contain only a few. This imbalance could negatively impact the effectiveness of contrastive learning. To address this, we first calculated the number of interactions per protein and the interquartile ranges (IQR) of these numbers. Specifically, the first quartile (Q1) was found to be 5, and the third quartile (Q3) was 129, resulting in an IQR of 124. To establish the upper bound, we added 1.5 times the IQR to Q3, arriving at a value of 315. Conversely, subtracting 1.5 times the IQR from Q1 yielded a negative value; hence, we manually selected a lower bound of 20. This manual adjustment aims to ensure more balanced protein representation in the dataset, with each protein having at least 10 positive and 10 negative interactions. Proteins with interaction counts outside these boundaries were excluded as outliers. In summary, we filtered the proteins to ensure a more balanced protein representation in the dataset, with each protein having a total

of at least 20 and at most 315 positive and negative interactions which shown in Figure 3.1.

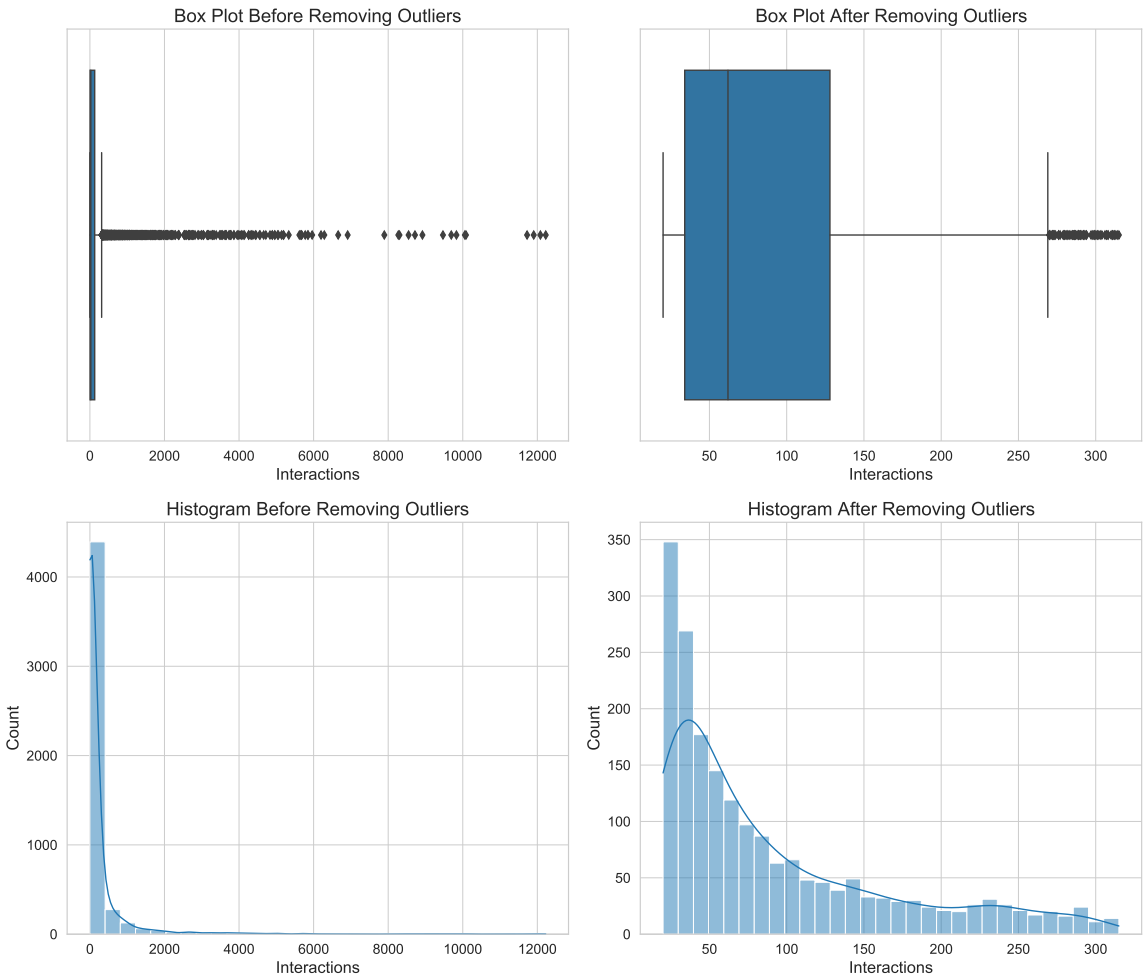

Figure 3.1. Boxplot analysis of the data before and after removing outliers.

Another important issue in creating the dataset was the similarity between interacting and non-interacting molecules, which could potentially deteriorate the effectiveness of contrastive learning training. To address this, we aim to represent interacting and non-interacting molecules for each protein with minimal similarities to enhance the effectiveness of contrastive learning. We compute the Tanimoto similarity between every pair of interacting and non-interacting molecules for each protein and discarded those with a Tanimoto similarity greater than 0.7, prioritizing the removal of non-interacting molecules. For data splitting, we calculate the average Tanimoto similarities of each molecule with oppositely labeled molecules and selected the 10 least

similar interacting and non-interacting molecules per protein. The subsequent 5 least similar interacting molecules in terms of average Tanimoto similarity are used for the test set, while the next consecutive 5 least similar molecules, both interacting and non-interacting, form the validation set. For a fair comparison with baseline models, we utilize an additional test set from [25]. This test set is specifically formed to create a diverse dataset, sampled from various Pfam families [46] and featuring different similarities. We exclude 50 proteins used in the training of this work and refer to this set as the unseen set. The summary of the data splits is presented in Table 3.1.

Table 3.1. Training, validation and test data summary.

Label	**Interactions**	**Unique Proteins**	**Unique Ligands**
Train	13137	657	11865
Validation	4039	404	3959
Test	3270	654	3205
Test Unseen	18871	150	18441

3.2. Model

We present ConGen, a targeted molecule design model that integrates pretrained protein and molecule encoders with contrastive learning and latent optimization techniques. ConGen is composed of three key components: 1) pretrained encoders for representing proteins and molecules, 2) projection layers that project protein and molecule representations into a shared latent space, and 3) a molecule decoder for generating novel compounds. The architecture of ConGen is illustrated in Figure 3.2. ConGen's training involves two stages: 1) a contrastive pretraining step, in which protein and molecule projectors are learned to map representations from different encoders into the same representation space, and 2) a latent optimization step, specific to the protein target of interest. In this step, a latent molecule representation is optimized to closely align with the representation of the target, starting from a random latent representation. In 3) molecule generation step, the target-specific latent representation is then fed into the molecule decoder to generate novel molecules. We utilize MoLeR's encoder

and decoder components as the molecule encoder and decoder, and ProtBERT as the protein encoder, with their weights frozen. ConGen is designed to be flexible and can integrate other pretrained protein and molecular models. The only requirement is that the molecule encoder and decoder must rely on the same latent space.

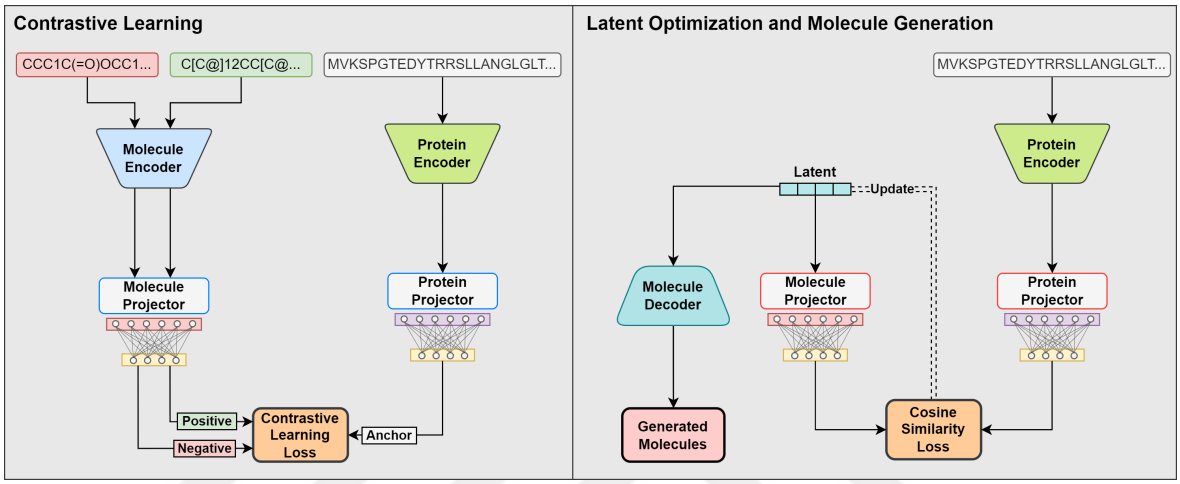

Figure 3.2. Model pipeline of ConGen.

3.2.1. Classification Task

The most straightforward task for our model is a classification task for drug-target interaction (DTI) predictions. During training, a protein sequence passes through the protein encoder and then the protein projector, while a molecule passes through the molecule encoder and then the molecule projector, as described in Figure 3.2. Then, utilizing the projections, we calculate the cosine similarity between two projections to predict the interaction between a molecule and a protein. Subsequently, the predicted label and the true label from the dataset are used to compute the binary cross entropy loss function. The goal is to accurately predict whether the given drug and target protein are interacting. Given y_i and $\hat{y}_i$ are the true label and the predicted probability for the i-th data point, the binary cross entropy loss function is the criterion on which the model is trained for drug-target interaction (DTI) prediction, described as

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right). \tag{3.1}$$

3.2.2. Contrastive Learning

The goal of the contrastive learning stage is to pull protein and interacting molecules pairs together while pushing non-interacting molecules apart, thereby creating a representation space where proteins are close to their interacting molecules and distant from those less likely to interact. Initial representations are obtained using the MoLeR encoder for molecules and ProtBERT for proteins. The protein representation is derived by averaging token embeddings, resulting in a 1024-dimensional representation, while the molecule representation, directly outputted from the MoLeR encoder, has a 512-dimensional vector. These representations are then independently passed through projection layers, mapping them into a shared space where the dimensions of the representations are unified at 512. This process effectively brings the vectors of interacting molecules and protein closer while distancing those of non-interacting molecules, as illustrated in Figure 3.3.

For contrastive learning, we first batch a protein along with its interacting and non-interacting molecules. In each batch, only one protein and its associated interacting and non-interacting molecules are processed. Triplet, N-pair and InfoNCE losses that are described in Section 2.5.1 are calculated. For triplet loss, the model operates on triplets consisting of an anchor, a positive sample, and a negative sample. In our context, anchor and positive can be either protein or interacting molecule while negative need to be chosen from non-interacting molecules. This process is same for N-pair loss but multiple non-interacting molecules are chosen as negatives for loss calculation. For InfoNCE, the approach is more like a binary classification task rather than using triplets. In this method, a protein and its interacting molecule are assigned with the same label, while non-interacting molecules receive a different label. During loss calculation, pairs of samples are selected. If a pair shares the same label, the pair is considered positive; if they have different labels, the pair is considered negative. The loss is computed for each positive pair in a batch, along with the negative pairs that share the same anchor.

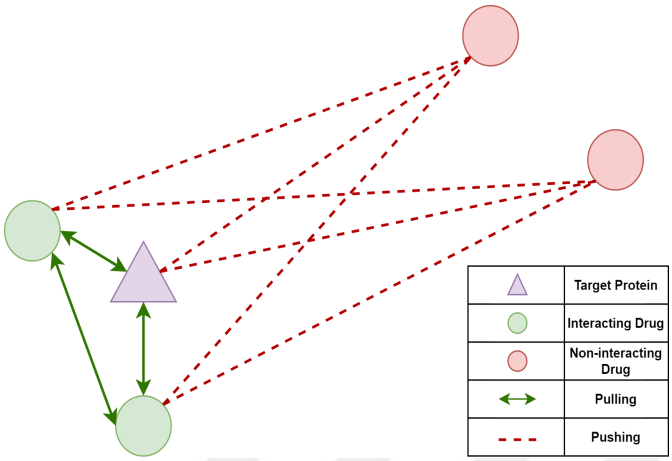

Figure 3.3. Illustration of contrastive learning.

3.2.3. Latent Optimization

The aim of the latent optimization stage is to optimize a randomly initialized representation towards a target protein of interest. A latent representation of size 512, which is compatible with the MoLeR representation space, is initialized randomly. It is then passed through molecule projection layers, mapping it into the same space as the target protein representation. The latter is obtained through the pretrained protein encoder and protein projection layers. It is important to note that we freeze the parameters of both the drug and target projectors, leaving the randomly initialized latent vector as the only learnable parameter. The optimization process involves adjusting the latent vector's projection values to closely align with the target protein projection. The process is summarized in Figure 3.4. For this purpose, the cosine similarity between the protein (p) and latent (l) projections is calculated and then multiplied by -1 to convert it into the cosine loss (L_{CosSim}) which is described as

$$L_{\text{CosSim}}(l, p) = -1 \cdot \frac{l \cdot p}{\max(\|l\|_2 \cdot \|p\|_2, \epsilon)}. \tag{3.2}$$

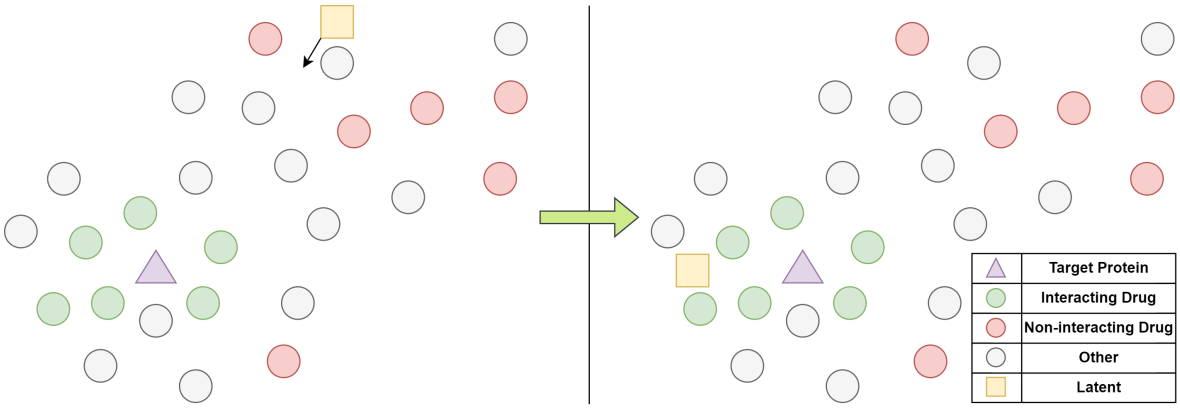

Figure 3.4. Illustration of latent optimization.

3.2.4. Molecule Generation

The target-optimized latent representation can then be used to generate target-specific molecules. By feeding this latent representation into the molecule decoder, new molecules can be generated, leveraging the latent's proximity to the target. Due to the decoder's limitation of generating only one molecule per vector, we employ a strategy to derive multiple molecules from a single latent vector by exploring its molecular "neighborhood". The concept of neighborhood refers to the variations of the same molecules, where the closest pairs of molecules are very similar in terms of chemical properties and scaffold similarity. Starting from an initial latent vector as the center, the neighborhood is formed and creating two normalized orthogonal vectors with uniform distribution. Next, the 4 directions of the central latent vector are examined in 5 steps by adding direction vectors and their negatives in 5 steps to the central latent vector. The challenge is to ensure that the added noise is significant enough to prompt the decoder to produce distinct molecules; otherwise, it might repetitively generate the same central molecule. Therefore, we sequentially traverse the generated molecules starting from the center and moving outward, adding a new molecule only if it is unique. An example of generated molecule neighborhood is given in Appendix A.1.

3.3. Implementation Details

ConGen was developed using PyTorch [47] and PyTorch Lightning [48], incorporating various libraries for its components. For the pretrained protein encoder, we employed the Hugging Face Transformers library [49]. For MoLeR, we used its official implementation[1] . During ConGen's contrastive learning phase, training spanned 50 epochs with checkpoints created at the end of each epoch. The checkpoint demonstrating the best validation performance was selected for further use. The training utilized a learning rate of 1E-4 with the AdamW optimizer [50]. The batches are composed of one protein along with 10 interacting and 10 non-interacting drug pairs, making a total of 21 items per batch. The learning rate was dynamically adjusted using a cosine annealing schedule with warm restarts [51]. In the latent optimization phase, each test protein's randomly initialized latent representation was trained over 100 epochs, using a learning rate of 0.01 and the Adam optimizer.

3.4. Evaluation

3.4.1. Precision and Recall

To evaluate the model's performance on DTI task, we utilized two metrics the average area under the precision–recall curve (AUPR) and the area under the receiver operating characteristic (AUROC). The ROC curve is obtained by plotting true positive rate (TPR) against false positive rate (FPR). On the other hand, precision-recall curve is obtained by plotting precision against recall. Given TP, FN and FP refer to true positives, false negatives and false positives, respectively, TPR, FPR and precision are defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{3.3}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{3.4}$$

[1] https://github.com/microsoft/molecule-generation

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3.5}$$

3.4.2. Metrics

We utilize a set of metrics from the MOSES benchmark [26] to evaluate the quality and diversity of generated molecules, as well as their similarity to reference molecules, specifically interacting molecules of the targets.

- Uniqueness (Uniq): The ratio of unique molecules.
- Internal diversity (IntDiv): The average one minus Tanimoto similarity within a set of compounds.
- Filters: Ratio of molecules that pass MOSES filters (MCF & PAINS).
- Frechet ChemNet Distance (FCD): A distance that quantifies how closely related two sets of molecules are, considering their chemical structure and biological properties [52].
- Single nearest neighbour (SNN): Average nearest neighbour similarity of valid and unique molecules to the reference dataset.
- Scaffold similarity (Scaf): Cosine similarity between the count vector of Bemis–Murcko scaffolds [53].
- Fragment similarity (Frag): Cosine similarity between the frequencies of BRICS fragments of two sets of molecules [54].
- The logarithm of the partition coefficient (logP): Compound's ability to dissolve in fats, oils, lipids, and non-polar solvents versus water, calculated as Wasserstein distance to reference distribution [55].
- Natural product-likeness (NP): How closely a molecule resembles compounds found in nature, calculated as Wasserstein distance to reference distribution [56].
- Synthetic Accessibility (SA): The ease of synthesizing a chemical compound, is calculated using Wasserstein distance to a reference distribution [57].
- Quantitative Estimate of Drug-likeness (QED): A quantity to estimate how likely a compound is to become an orally active drug in humans, calculated as Wasser-

stein distance to reference distribution [58].

3.4.3. t-Distributed Stochastic Neighbor Embedding Visualisation

The generated latent space is visualized using t-Distributed Stochastic Neighbor Embedding t-SNE representation [59, 60], where projection vectors are transformed into two-dimensional data points for analysis. We visualised projections of the protein, interacting, non-interacting and randomly selected molecules. We aimed to show how contrastive learning strategy affects the shared space by comparing a randomly initialized model and a trained model.

In the next visualization, we have included proteins, interacting and non-interacting molecules, along with both randomly initialized and optimized latent vectors. The purpose of this visualization is to demonstrate how latent optimization moves randomly initialized latents closer to the target protein and interacting molecules.

4. RESULTS

4.1. ConGen

4.1.1. DTI Prediction Task.

We assessed the performance of the model for DTI prediction task. While keeping other parameters and the architecture components constant, we changed the molecule encoder and experimented three different molecule encoders: ChemBERTa, MolT5, and MegaMolBART. In Table 4.1, The results for AUPR and AUROC using these encoders are reported and compared with ConPLex which utilizes Morgan Fingerprint representations for encoding molecules. In terms of AUPR, MegaMolBART appears to have slightly better results compared to the other representation methods. On the other hand, for AUROC, the results are nearly identical across the four different representation methods. This indicates that the chosen encoding methods do not have a major effect on the success of the binary classification task.

Table 4.1. AUROC and AUPR results for different encoding methods.

Encoder	**AUPR**	**AUROC**
ConPLex	0.699	0.926
ChemBERTa	0.681	0.922
MolT5	0.687	0.922
MegaMolBART	0.704	0.928

In Figure 4.1, the cosine similarity between each protein and its interacting molecule pairs is calculated. Subsequently, the calculated cosine similarities are averaged for each protein. The same process is repeated for non-interacting molecules for each protein, as described in 4.2. The expectation is that the average cosine similarities for interacting pairs will be closer to 1, indicating strong similarity, while those for non-interacting pairs will be closer to 0, indicating little or no similarity. The clearer the separation between the average cosine similarities of interacting and non-interacting

pairs, the better the performance of the encoding method. While there is an observable separation for each encoding method, the differences between the encoding methods are not significant. This suggests that the choice of encoding method alone does not significantly enhance the separation.

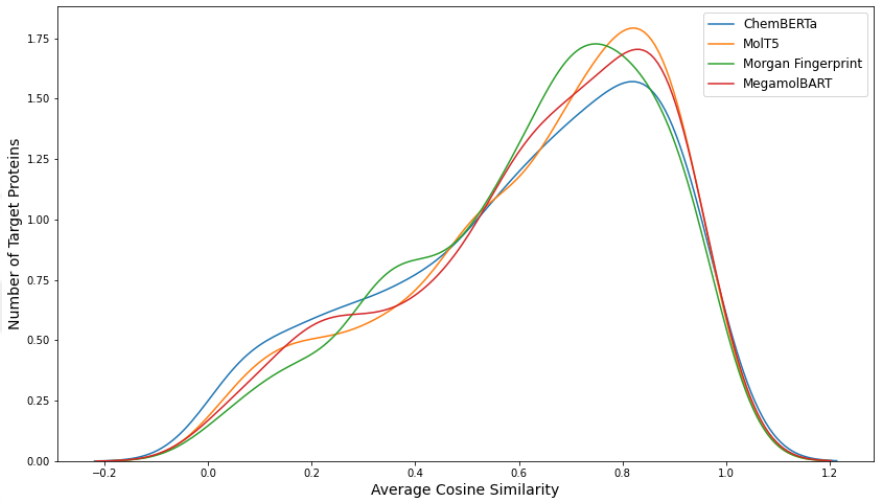

Figure 4.1. Average cosine similarities between interacting molecules and proteins.

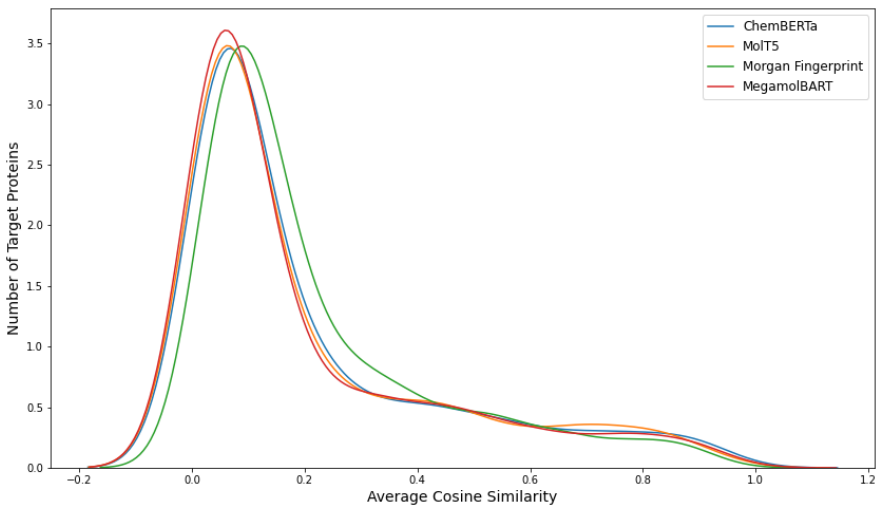

Figure 4.2. Average cosine similarities between non-interacting molecules and proteins.

4.1.2. Loss Function Comparison.

We conducted three distinct experiment for comparing different losses: triplet, InfoNCE and N-pair loss. Upon analyzing the results, we observed that for metrics such as IntDiv, Scaf, Frag, and SNN, N-pair loss slightly outperforms the others. However, for FCD, InfoNCE shows a significantly lower value. Figure 4.3 illustrates the distribution of FCD scores across proteins for each model with the specified loss function. The boxplot results show that both InfoNCE and triplet yield lower FCD scores compared to N-pair loss. Triplet loss yields lowest minimum and maximum score for FCD and is comparable in average score to InfoNCE. These results indicate that both models with triplet loss and InfoNCE loss performs better than model trained with N-pair loss.

Table 4.2. Comparison of different loss functions on model's performance.

Model	**IntDiv ↑**	**Filters ↑**	**Scaf ↑**	**Frag ↑**	**SNN ↑**	**FCD ↓**
Triplet	0.806	0.813	0.000	0.566	0.313	16.771
N-pair	0.840	0.881	0.010	0.905	0.327	16.276
InfoNCE	0.818	0.881	0.002	0.846	0.321	13.465

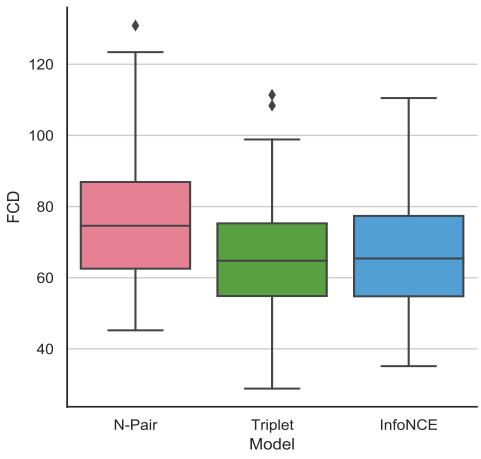

Figure 4.3. A comparative analysis of ConGen model with different losses in terms of FCD per target distribution. Lower FCD is better.

4.1.3. Ablation Study

We conducted an ablation study to assess the effectiveness of ConGen's contrastive learning and latent optimization stages. This involved comparing ConGen with two different setups on the test set developed for this study. ConGen model is trained using the InfoNCE loss.

The first setup excluded the contrastive learning stage. Here, protein and molecule embeddings obtained from pretrained models were projected into a space using randomly initialized projection layers and then subjected to the latent optimization stage. This resulted in random latent initialization. The purpose of this setup was to demonstrate the role of contrastive learning in aligning interacting molecules closer to proteins while distancing non-interacting molecules. We refer to this setup as "without contrastive learning (CL)".

The second setup lacked both contrastive learning (CL) and latent optimization (LO) stages. Instead, it involved directly generating molecules from randomly generated latent vectors using the MoLeR decoder. Although MoLeR is not specifically designed for targeted drug design, its training on the GuacoMol dataset, which contains approximately 1.5M drug-like molecules [31, 61], makes it a valuable baseline and a pertinent part of our ablation study. We refer to this setup as "without contrastive learning (CL) and latent optimization (LO)".

The results of the ablation study, presented in Table 4.3, demonstrate the effectiveness of contrastive learning and combining it with latent optimization. We observed that ConGen achieved the lowest FCD score, followed by the setup without contrastive learning, and then the setup without both contrastive learning and latent optimization. Furthermore, ConGen showed better or equivalent uniqueness compared to the ablation experiments. This trend continued for another similarity metric, Frag, but the performance slightly worsens for the remaining metrics.

Table 4.3. Ablation study results.

Model	**IntDiv ↑**	**Filters ↑**	**Scaf ↑**	**Frag ↑**	**SNN ↑**	**FCD ↓**
ConGen	0.824	0.887	0.0137	0.863	0.299	10.474
Without CL	0.810	0.922	0.0255	0.622	0.307	17.082
Without CL&LO	0.802	0.949	0.0102	0.586	0.322	21.437

4.1.4. Baseline Comparison

We trained ConGen$_I$ with InfoNCE loss and ConGen$_T$ with triplet loss. We compared trained models with two baseline models: MoLeR, a generic molecule generator not specific to any target, and EncDecLM, a state-of-the-art sequence-based target-specific molecule generator, on the unseen set. A maximum of 20 different molecules are generated for each protein in the unseen test dataset using every model listed in Table 4.4. MoLeR, being an autoencoder trained to reconstruct the original molecule, is not capable of generating target-specific molecules. For molecule generation with MoLeR, random latent representations were fed into the model, positioning MoLeR as a baseline to assess ConGen's generic molecule generative abilities.

The quality and diversity of the generated molecules were then evaluated using metrics from the MOSES benchmark and molecular property distance metrics (logP, NP, SA, QED). These metrics were computed at the set level, aggregating all generated molecules for comparison against all reference molecules. Lastly, FCD results are analyzed at the protein level, considering the generated molecules and the interacted molecules of each protein separately.

The MOSES score results are shown in Table 4.4 report the at the set level while the molecular distance metrics are analyzed in Table 4.5. They reveal that both ConGen models outperformed MoLeR and EncDecLM on many metrics assessing the feasibility of the generated molecules (e.g., Uniq, logP, NP, SA, QED), highlighting the quality of the molecules generated by ConGen. ConGen also achieved a lower FCD score, which measures the distance of generated molecules to interacting molecules,

than MoLeR, indicating its target specificity. However, EncDecLM performed better than ConGen models in a set of similarity metrics (i.e., FCD, SNN, Scaf, Frag).

Table 4.4. Comparison between ConGen and baseline models for MOSES metrics.

Model	**Uniq ↑**	**IntDiv ↑**	**Filters ↑**	**Scaf ↑**	**Frag ↑**	**SNN ↑**	**FCD ↓**
ConGen_I	0.998	0.818	0.881	0.002	0.846	0.321	13.464
ConGen_T	0.976	0.806	0.813	0.000	0.566	0.313	16.771
MoLeR	0.947	0.800	0.959	0.001	0.520	0.345	24.383
EncDecLM	0.821	0.867	0.690	0.069	0.966	0.560	11.170

Figure 4.4 illustrates the distribution of FCD scores across proteins in the unseen dataset for each model. The results show that both ConGen models achieved, on average, a lower FCD score compared to MoLeR and maintained a mean FCD score comparable to that of EncDecLM. Notably, EncDecLM achieved a lower minimum score and first quartile. We suggest that this difference can be attributed to the size of the training datasets: EncDecLM was trained on a dataset comprising 310K interactions, whereas ConGen was trained on a significantly smaller dataset of only 13K interactions.

4.1.5. Space Visualization

We visualized the shared space by projecting a protein and its interacting and non-interacting molecules, along with random molecules using t-SNE representation [59].

Table 4.5. Comparison between ConGen and baseline models for molecular property distance metrics.

Model	**logP ↓**	**NP ↓**	**SA ↓**	**QED ↓**
ConGen_I	0.538	0.233	0.198	0.076
ConGen_T	0.393	0.328	0.340	0.083
MoLeR	0.757	0.441	0.757	0.235
EncDecLM	1.202	0.465	0.180	0.031

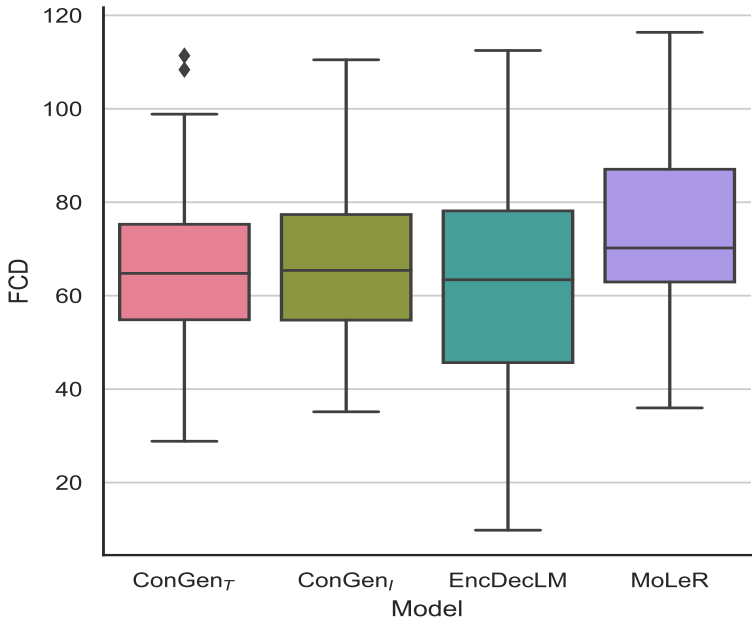

Figure 4.4. A comparative analysis of ConGen model with baseline models in terms of FCD per target distribution. Lower FCD is better.

Figure 4.5 illustrates that, before applying contrastive learning, the molecules and protein are randomly scattered throughout the space for protein with UniProt ID Q8CA95. In this figure, a comparison with and without contrastive learning is demonstrated. 'Others' representing a set of 200 molecules randomly selected from the dataset. In contrast, after applying contrastive learning, the projections reveal a more organized space where interacting molecules cluster around the target protein, while non-interacting molecules remain distant. This distribution facilitates targeted molecule generation. To confirm that these observations are not isolated to a single protein, we have conducted and provided additional experiments in Appendix B.

We also visualized the effect of latent optimization Figure 4.6. We observe space which is formed after contrastive learning with interacting molecules is clustered around target while non-interacting molecules are positioned at a distance. In this figure, 'Latent Initially' denotes the initial position of the latent variable, while 'Latent Final' indicates its position at the end of the optimization process. A randomly initialized latent vector may be anywhere within this space. However, with latent optimization, we observe that the final latent vector is positioned near the target and interacting drug molecules. By utilizing optimized latent vector, we can proceed to the molecule generation step, allowing for the production of more specifically targeted molecules for the designated protein. To support these results, several proteins are analyzed in Appendix C.

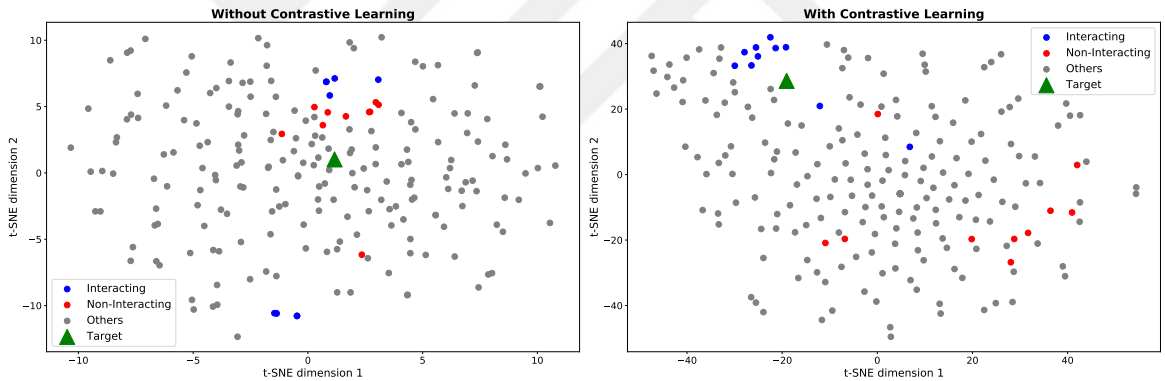

Figure 4.5. Visualization of the shared space before and after contrastive learning for protein with UniProt ID Q8CA95.

4.2. Web Service

Besides study of ConGen, we have developed a web service that can be used to deploy targeted molecule generative models. Some molecule generative models are hosted on network-based server available on the Huggingface website [49]. The server asks from the user to select the desired generator model. Currently, the only options are to select one of the two versions of a model: WarmMolGenOne or WarmMolGenTwo [25]. However, the system is designed to integrate additional models in the future, meaning it will allow for the inclusion of new models over time. Thus, the server has

the potential to become a significant tool for target-based molecule design models in the future.

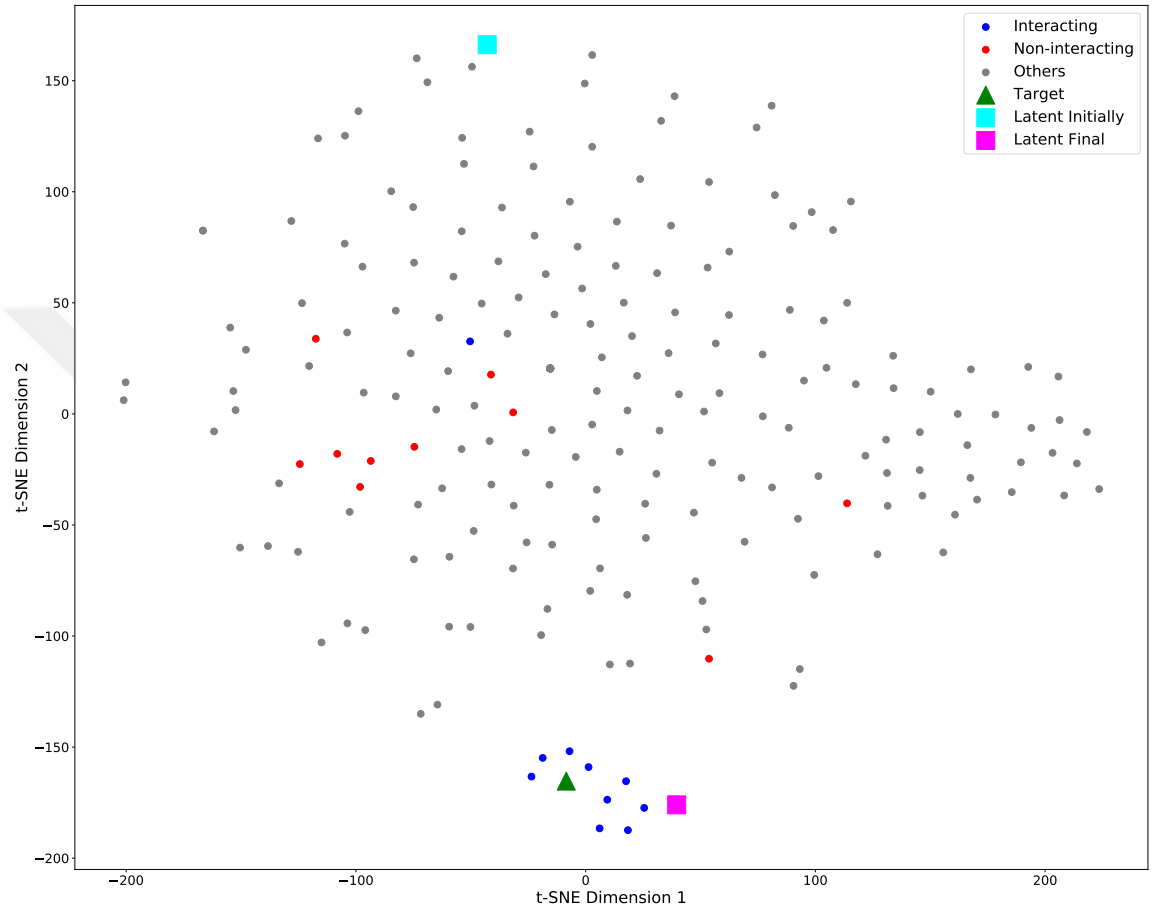

Figure 4.6. Visualization of latent space adjustment for protein UniProt ID P0A725.

Next, the user must specify the number of drug-like molecules they wish to produce. Since identifying lead molecules in drug design is of critical importance, the number specified by the user represents the desired quantity of lead molecules to produce. The default number is set to 10 when the site is initialized.

The maximum sequence length of the molecule to be produced must also be specified. Moreover, it is up to the user to decide whether or not sampling will be performed. Subsequently, the most crucial information that the user must enter is the target protein sequence. Without this information, a request to produce drug-like molecules from the server cannot be executed because the goal is targeted drug generation. Once the information is provided, molecules are generated by clicking the "Run" button at the bottom.

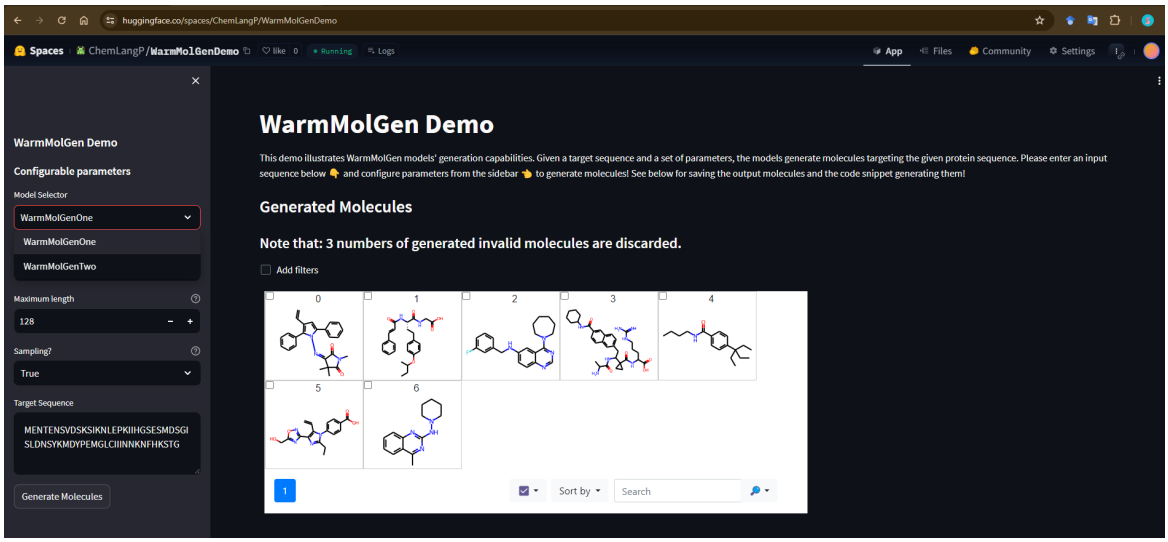

Figure 4.7. Web service.

The molecules are displayed on the screen in a 2-dimensional representation. Additionally, when hovered over with the cursor, their SMILES sequences can also be seen. Alongside SMILES sequences, criteria such as drug-likeness, molecular weight, synthesizability, logP, and natural product score are also provided to users as outputs. Based on this information, users can perform filtering and ranking processes. If the model produces a molecule that does not actually exist, this molecule is not presented to the user, but the number of such molecules generated and deleted under these conditions is reported to the user. 4.7 shows an example view of the network server we have described. This image can be shown as the result of our work in this context. Access to the network-based server can be obtained using the link [2] .

[2]https://huggingface.co/spaces/ChemLangP/WarmMolGenDemo

5. DISCUSSION

Our results suggest that the architecture of ConGen, which projects molecules and proteins into the same space, can be easily adapted to the drug target interaction task by simply changing the loss function to a classification loss. Moreover, it is observed that ConGen is effective in targeted molecule generation. ConGen is also a lightweight model, leveraging pretrained models and training a small number of parameters belonging to projection layers and latent representations. Another strength of the model is its flexibility. ConGen can be applied with different protein encoders and generative molecular models, as long as the molecular encoder and decoder rely on the same latent space.

Despite its potential contribution, ConGen has critical limitations. The performance of the model depends on the representativeness of the interaction data between proteins and ligands, which means that some proteins, molecules, or interactions thereof may not be adequately captured. Therefore, training a model with a more representative and larger amount of data is one of the areas for improvement in this study. While contrastive learning improves the model's ability to distinguish between interacting and non-interacting pairs, it requires careful selection of training and validation data. In this study, we devise a simple strategy that takes into account the Tanimoto similarity of the molecules, choosing the least similar ones, though this approach risks discarding some pairs that could be valuable. To enable ConGen to capture a diverse set of proteins effectively, a broader and more generalized dataset is required.

As future work, we plan to use various protein encoders and generative molecular models to assess the effectiveness of ConGen across different models. We also aim to train a larger model with a more complex and diversified dataset to improve performance. Enlarging the dataset and model size might provide a more generalized projection for target-ligand pairs. Another avenue of research could be extending this framework to multiple targets, designing a new loss that encourages the model to

generate molecules targeting specific targets while discouraging targeting others.

6. CONCLUSION

In this study, we initially utilized an encoding and projection architecture for the drug-target interaction (DTI) task, applying it as a binary classification function. We compared different encoding methods for molecule encoding in the drug-target interaction (DTI) task, achieving results that are competitive with state-of-the-art models. Additionally, the variations in the encoding methods experimented did not significantly affect the results. Then, we introduced ConGen, a novel model that employs contrastive learning to generate a shared space for protein-ligand pairs and utilizes latent optimization to generate molecules specific to target proteins. We investigated the performance of different contrastive learning loss functions resulting InfoNCE and triplet loss are best performing ones. Moreover, we compared our model's performance against state-of-the-art works on an unseen test set, where our model generally outperformed MoLeR but fell behind EncDecLM. We also performed an ablation study, demonstrating the effectiveness of the contrastive learning and latent optimization in our model. As the final step of this thesis, a web service has been developed that can deploy targeted molecule generative models. This service is capable of deploying any sequence-based model that takes a target protein sequence as input and outputs molecules, including the ConGen model.

REFERENCES

1. Bohacek, R. S., C. McMartin and W. C. Guida, "The Art and Practice of Structure-Based Drug Design: A Molecular Modeling Perspective", *Medicinal Research Reviews*, Vol. 16, No. 1, pp. 3–50, 1996.

2. DiMasi, J. A., L. Feldman, A. Seckler and A. Wilson, "Trends in Risks Associated With New Drug Development: Success Rates for Investigational Drugs", *Clinical Pharmacology & Therapeutics*, Vol. 87, No. 3, p. 272–277, 2010.

3. Hughes, J., S. Rees, S. Kalindjian and K. Philpott, "Principles of Early Drug Discovery", *British Journal of Pharmacology*, Vol. 162, No. 6, p. 1239–1249, 2011.

4. Shaker, B., S. Ahmad, J. Lee, C. Jung and D. Na, "In Silico Methods and Tools for Drug Discovery", *Computers in Biology and Medicine*, Vol. 137, p. 104851, 2021.

5. Eckmann, P., K. Sun, B. Zhao, M. Feng, M. K. Gilson and R. Yu, "LIMO: Latent Inceptionism for Targeted Molecule Generation", K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu and S. Sabato (Editors), *International Conference on Machine Learning*, Vol. 162, pp. 5777–5792, Baltimore, Maryland, USA, 2022.

6. Krishnan, S. R., N. Bung, S. R. Vangala, R. Srinivasan, G. Bulusu and A. Roy, "De Novo Structure-based Drug Design Using Deep Learning", *Journal of Chemical Information and Modeling*, Vol. 62, No. 21, p. 5100–5109, 2021.

7. Li, Y., P. H. da Costa Avelar, X. Chen, L. Zhang, M. Wu and S. Tsoka, "CLigOpt: Controllable Ligand Design through Target-Specific Optimisation", BioRxiv 2024.03.15.585255, 2024.

8. Telepov, A., A. Tsypin, K. Khrabrov, S. Yakukhnov, P. Strashnov, P. Zhilyaev, E. Rumiantsev, D. Ezhov, M. Avetisian, O. Popova *et al.*, "FREED++: Improving RL Agents for Fragment-Based Molecule Generation by Thorough Reproduction",

ArXiv 2401.09840, 2024.

9. Yang, S., D. Hwang, S. Lee, S. Ryu and S. J. Hwang, "Hit and Lead Discovery with Explorative RL and Fragment-based Molecule Generation", M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang and J. W. Vaughan (Editors), *Advances in Neural Information Processing Systems*, Vol. 34, pp. 7924–7936, Curran Associates, Inc., Online, 2021.

10. Wen, M., Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun and H. Lu, "Deep-Learning-based Drug–Target Interaction Prediction", *Journal of Proteome Research*, Vol. 16, No. 4, p. 1401–1409, 2017.

11. Öztürk, H., A. Özgür and E. Ozkirimli, "DeepDTA: Deep Drug–Target Binding Affinity Prediction", *Bioinformatics*, Vol. 34, No. 17, p. i821–i829, 2018.

12. Huang, K., C. Xiao, L. M. Glass and J. Sun, "MolTrans: Molecular Interaction Transformer for Drug–Target Interaction Prediction", *Bioinformatics*, Vol. 37, No. 6, p. 830–836, 2020.

13. Svensson, E., P.-J. Hoedt, S. Hochreiter and G. Klambauer, "HyperPCM: Robust Task-Conditioned Modeling of Drug–Target Interactions", *Journal of Chemical Information and Modeling*, Vol. 64, No. 7, p. 2539–2553, 2024.

14. NVIDIA, "MegaMolBART", 2021, `https://github.com/NVIDIA/MegaMolBART`, accessed on December 20, 2023.

15. Chithrananda, S., G. Grand and B. Ramsundar, "ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction", ArXiv 2010.09885, 2020.

16. Edwards, C., T. Lai, K. Ros, G. Honke, K. Cho and H. Ji, "Translation between Molecules and Natural Language", , 2022, arXiv 2204.11817, 2022.

17. Luo, S., J. Guan, J. Ma and J. Peng, "A 3D Generative Model for Structure-Based Drug Design", M. Ranzato *et al.* (Editors), *Advances in Neural Information Processing Systems*, Vol. 34, pp. 6229–6239, Online, 2021.

18. Liu, M., Y. Luo, K. Uchino, K. Maruhashi and S. Ji, "Generating 3D Molecules for Target Protein Binding", K. Chaudhuri *et al.* (Editors), *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162, pp. 13912–13924, Baltimore, Maryland, USA, 2022.

19. Peng, X., S. Luo, J. Guan, Q. Xie, J. Peng and J. Ma, "Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets", K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu and S. Sabato (Editors), *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162, pp. 17644–17655, Baltimore, Maryland, USA, 2022.

20. Schneuing, A., Y. Du, C. Harris, A. Jamasb, I. Igashov, W. Du, T. Blundell, P. Lió, C. Gomes, M. Welling, M. Bronstein and B. Correia, "Structure-Based Drug Design with Equivariant Diffusion Models", ArXiv 2210.13695, 2023.

21. Buttenschoen, M., G. M. Morris and C. M. Deane, "PoseBusters: AI-Based Docking Methods Fail to Generate Physically Valid Poses or Generalise to Novel Sequences", *Chemical Science*, Vol. 15, No. 9, p. 3130–3139, 2024.

22. Harris, C., K. Didi, A. Jamasb, C. Joshi, S. Mathis, P. Lio and T. Blundell, "PoseCheck: Generative Models for 3D Structure-Based Drug Design Produce Unrealistic Poses", *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, New Orleans, USA, 2023.

23. 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*, Vol. 32-33, pp. 55–63, 2019.

24. Grechishnikova, D., "Transformer Neural Network for Protein-Specific de Novo Drug Generation as a Machine Translation Problem", *Scientific Reports*, Vol. 11, No. 1, p. 321, 2021.

25. Uludoğan, G., E. Ozkirimli, K. O. Ulgen, N. Karalı and A. Özgür, "Exploiting Pretrained Biochemical Language Models for Targeted Drug Design", *Bioinformatics*, Vol. 38, No. Supplement_2, pp. ii155–ii161, 2022.

26. Polykovskiy, D., A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov *et al.*, "Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models", *Frontiers in Pharmacology*, Vol. 11, pp. 1–10, 2020.

27. Qian, H., C. Lin, D. Zhao, S. Tu and L. Xu, "Alphadrug: Protein Target Specific de Novo Molecular Generation", *PNAS Nexus*, Vol. 1, No. 4, p. pgac227, 2022.

28. Jaiswal, A., A. R. Babu, M. Z. Zadeh, D. Banerjee and F. Makedon, "A Survey on Contrastive Self-Supervised Learning", *Technologies*, Vol. 9, No. 1, p. 2, 2020.

29. Singh, R., S. Sledzieski, B. Bryson, L. Cowen and B. Berger, "Contrastive Learning in Protein Language Space Predicts Interactions Between Drugs and Protein Targets", *Proceedings of the National Academy of Sciences*, Vol. 120, No. 24, p. e2220778120, 2023.

30. Liu, S., W. Nie, C. Wang, J. Lu, Z. Qiao, L. Liu, J. Tang, C. Xiao and A. Anandkumar, "Multi-Modal Molecule Structure-Text Model for Text-Based Retrieval and Editing", *Nature Machine Intelligence*, Vol. 5, No. 12, pp. 1447–1457, 2023.

31. Maziarz, K., H. R. Jackson-Flux, P. Cameron, F. Sirockin, N. Schneider, N. Stiefl, M. Segler and M. Brockschmidt, "Learning to Extend Molecular Scaffolds with Structural Motifs", *International Conference on Learning Representations*, Online, 2022.

32. Elnaggar, A., M. Heinzinger, C. Dallago, G. Rehawi, 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", BioRxiv 2020.07.12.199554, 2020.

33. Tonkens, R., "An Overview of the Drug Development Process", *Physician Executive*, Vol. 31, No. 3, pp. 48–52, 2005.

34. Alberts, B., A. Johnson, J. Lewis, P. Walter, M. Raff and K. Roberts, *Molecular Biology of the Cell*, Routledge, San Francisco, 4th edition, international student edition edn., 2002.

35. Bowie, J. U., R. Lüthy and D. Eisenberg, "A Method to Identify Protein Sequences That Fold into a Known Three-Dimensional Structure", *Science*, Vol. 253, No. 5016, pp. 164–170, 1991.

36. Chopra, S., R. Hadsell and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 539–546 vol. 1, San Diego, California, 2005.

37. Schroff, F., D. Kalenichenko and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", ArXiv 1503.03832, 2015.

38. Sohn, K., "Improved Deep Metric Learning with Multi-class N-pair Loss Objective", D. Lee *et al.* (Editors), *Advances in Neural Information Processing Systems*, Vol. 29, Curran Associates, Inc., Barcelona, Spain, 2016.

39. van den Oord, A., Y. Li and O. Vinyals, "Representation Learning with Contrastive Predictive Coding", ArXiv 1807.03748, 2018.

40. Gutmann, M. and A. Hyvärinen, "Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models", Y. W. Teh and M. Tittering-

ton (Editors), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9, pp. 297–304, Sardinia, Italy, 2010.

41. Bank, D., N. Koenigstein and R. Giryes, "Autoencoders", ArXiv 2003.05991, 2020.

42. Sanchez-Lengeling, B., E. Reif, A. Pearce and A. Wiltschko, "A Gentle Introduction to Graph Neural Networks", *Distill*, Vol. 6, No. 8, 2021.

43. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention Is All You Need", I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (Editors), *Advances in Neural Information Processing Systems*, Vol. 30, Long Beach, CA, USA, 2017.

44. 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, p. D1045–D1053, Oct. 2015.

45. Gao, K. Y., A. Fokoue, H. Luo, A. Iyengar, S. Dey and P. Zhang, "Interpretable Drug Target Prediction Using Deep Neural Representation", *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3371–3377, Stockholm, Sweden, 2018.

46. Mistry, J., S. Chuguransky, L. Williams, M. Qureshi, G. A. Salazar, E. L. L. Sonnhammer, S. C. E. Tosatto, L. Paladin, S. Raj, L. J. Richardson *et al.*, "Pfam: The Protein Families Database in 2021", *Nucleic Acids Research*, Vol. 49, No. D1, p. D412–D419, 2020.

47. Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", ArXiv 1912.01703, 2019.

48. Falcon, W. and The PyTorch Lightning Team, "PyTorch Lightning", 2019, `https://github.com/Lightning-AI/lightning`, accessed on April 10, 2024.

49. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-Art Natural Language Processing", Q. Liu and D. Schlangen (Editors), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, 2020.

50. Loshchilov, I. and F. Hutter, "Decoupled Weight Decay Regularization", ArXiv 1711.05101, 2019.

51. Loshchilov, I. and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts", ArXiv 1608.03983, 2017.

52. 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, p. 1736–1741, 2018.

53. Bemis, G. W. and M. A. Murcko, "The Properties of Known Drugs. 1. Molecular Frameworks", *Journal of Medicinal Chemistry*, Vol. 39, No. 15, p. 2887–2893, 1996.

54. Degen, J., C. Wegscheid-Gerlach, A. Zaliani and M. Rarey, "On the Art of Compiling and Using "Drug-Like" Chemical Fragment Spaces", *ChemMedChem*, Vol. 3, No. 10, p. 1503–1507, 2008.

55. Kwon, Y., *Partition and Distribution Coefficients*, Springer, New York, NY, 2013.

56. Ertl, P., S. Roggo and A. Schuffenhauer, "Natural Product-likeness Score and Its Application for Prioritization of Compound Libraries", *Journal of Chemical Information and Modeling*, Vol. 48, No. 1, pp. 68–74, 2008.

57. Ertl, P. and A. Schuffenhauer, "Estimation of Synthetic Accessibility Score of Drug-like Molecules Based on Molecular Complexity and Fragment Contributions", *Journal of Cheminformatics*, Vol. 1, No. 1, p. 8, 2009.

58. Bickerton, G. R., G. V. Paolini, J. Besnard, S. Muresan and A. L. Hopkins, "Quantifying the Chemical Beauty of Drugs", *Nature Chemistry*, Vol. 4, No. 2, p. 90–98, 2012.

59. van der Maaten, L. and G. Hinton, "Visualizing Data Using t-SNE", *Journal of Machine Learning Research*, Vol. 9, No. 86, pp. 2579–2605, 2008.

60. Hinton, G. E. and S. Roweis, "Stochastic Neighbor Embedding", S. Becker, S. Thrun and K. Obermayer (Editors), *Advances in Neural Information Processing Systems*, Vol. 15, MIT Press, Vancouver, British Columbia, Canada, 2002.

61. Brown, N., M. Fiscato, M. H. Segler and A. C. Vaucher, "GuacaMol: Benchmarking Models for de Novo Molecular Design", *Journal of Chemical Information and Modeling*, Vol. 59, No. 3, p. 1096–1108, 2019.

APPENDIX A: DECODING STRATEGY

This section visualizes decoding strategy.

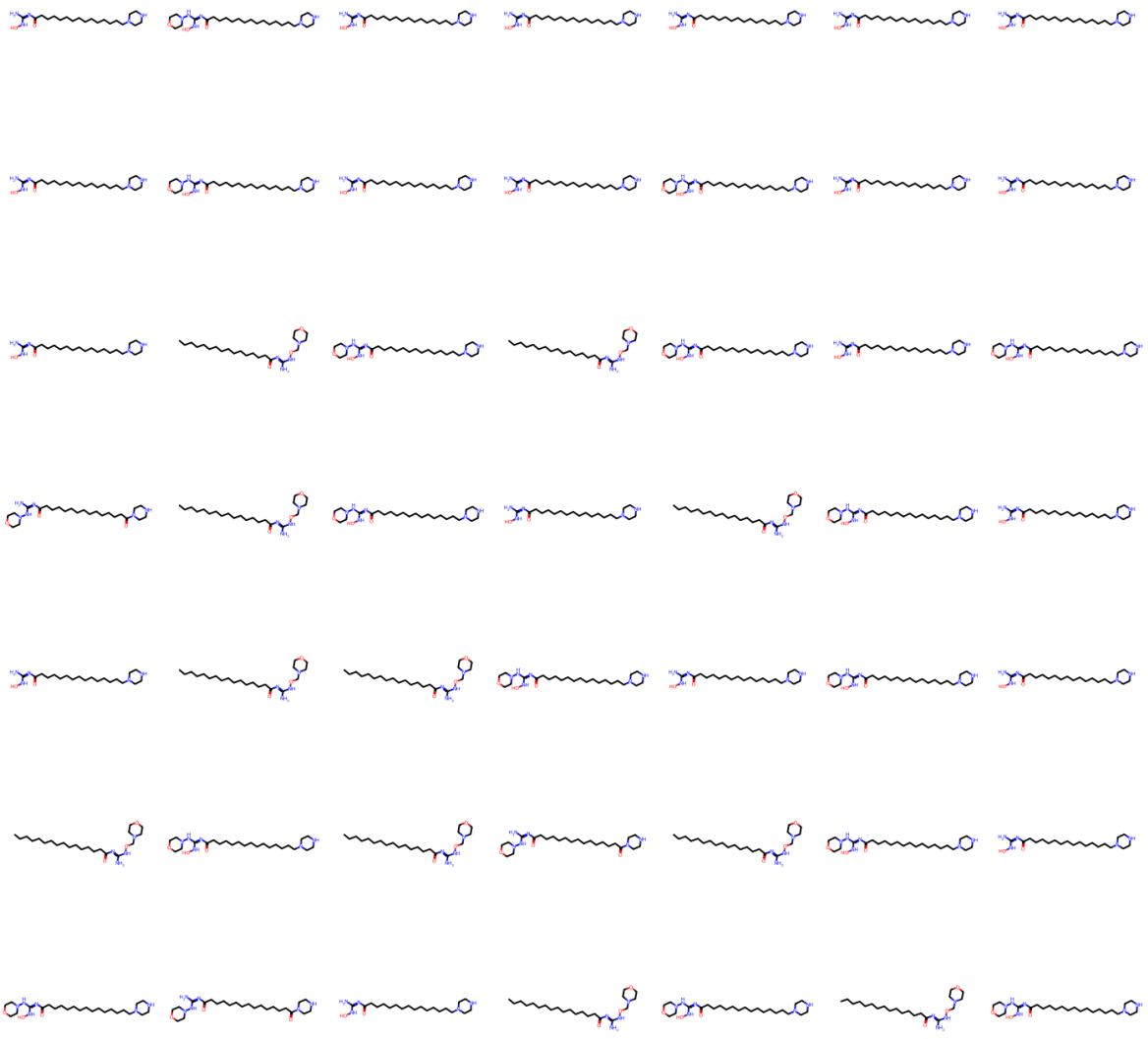

Figure A.1. Neighborhood of generated molecules.

APPENDIX B: CONTRASTIVE LEARNING T-SNE RESULTS

This section provides additional examples for contrastive learning t-SNE results.

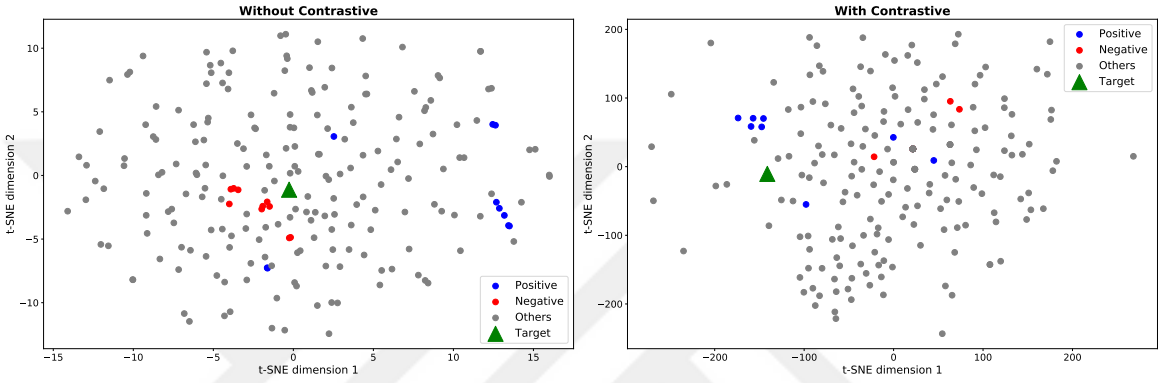

Figure B.1. Protein with UniProtID P97629.

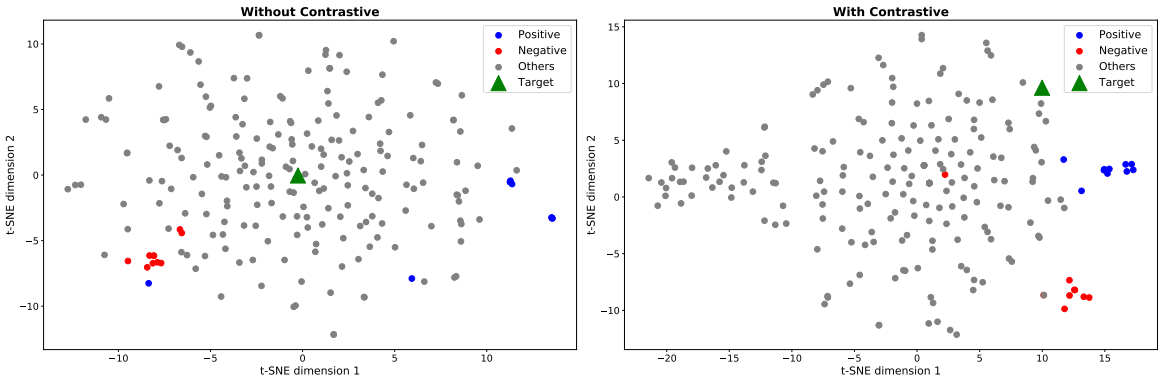

Figure B.2. Protein with UniProtID Q6GG09.

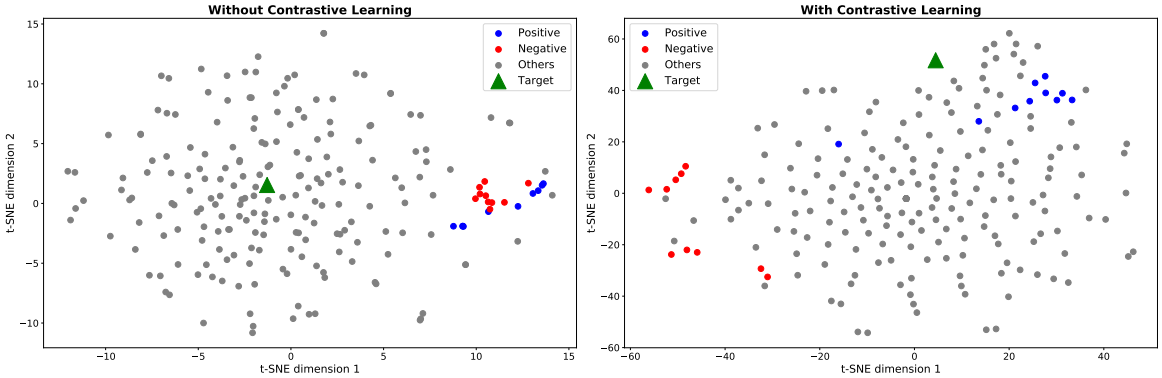

Figure B.3. Protein with UniProtID Q9Y6K0.

APPENDIX C: LATENT OPTIMIZATION T-SNE RESULTS

This section provides additional examples for latent optimization t-SNE results.

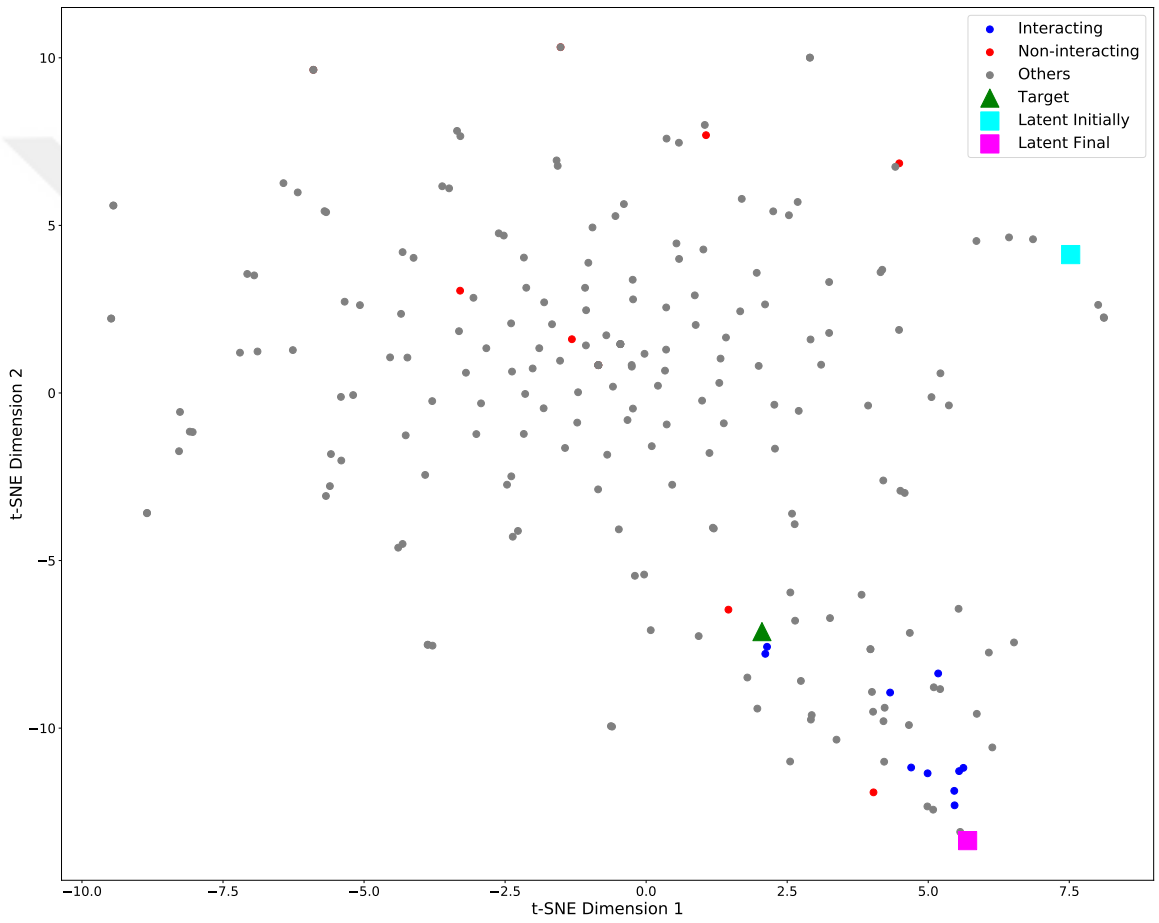

Figure C.1. Protein with UniProtID P09769.

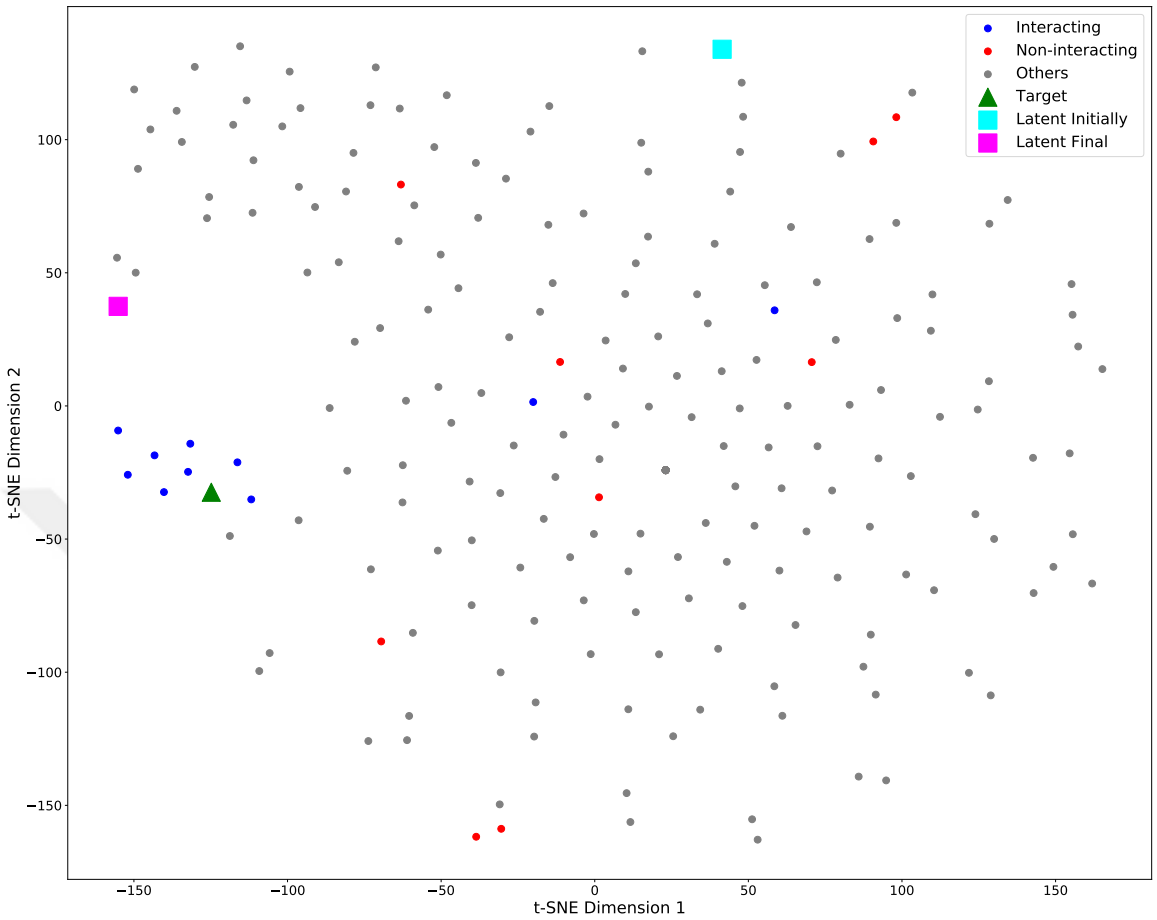

Figure C.2. Protein with UniProtID P36896.

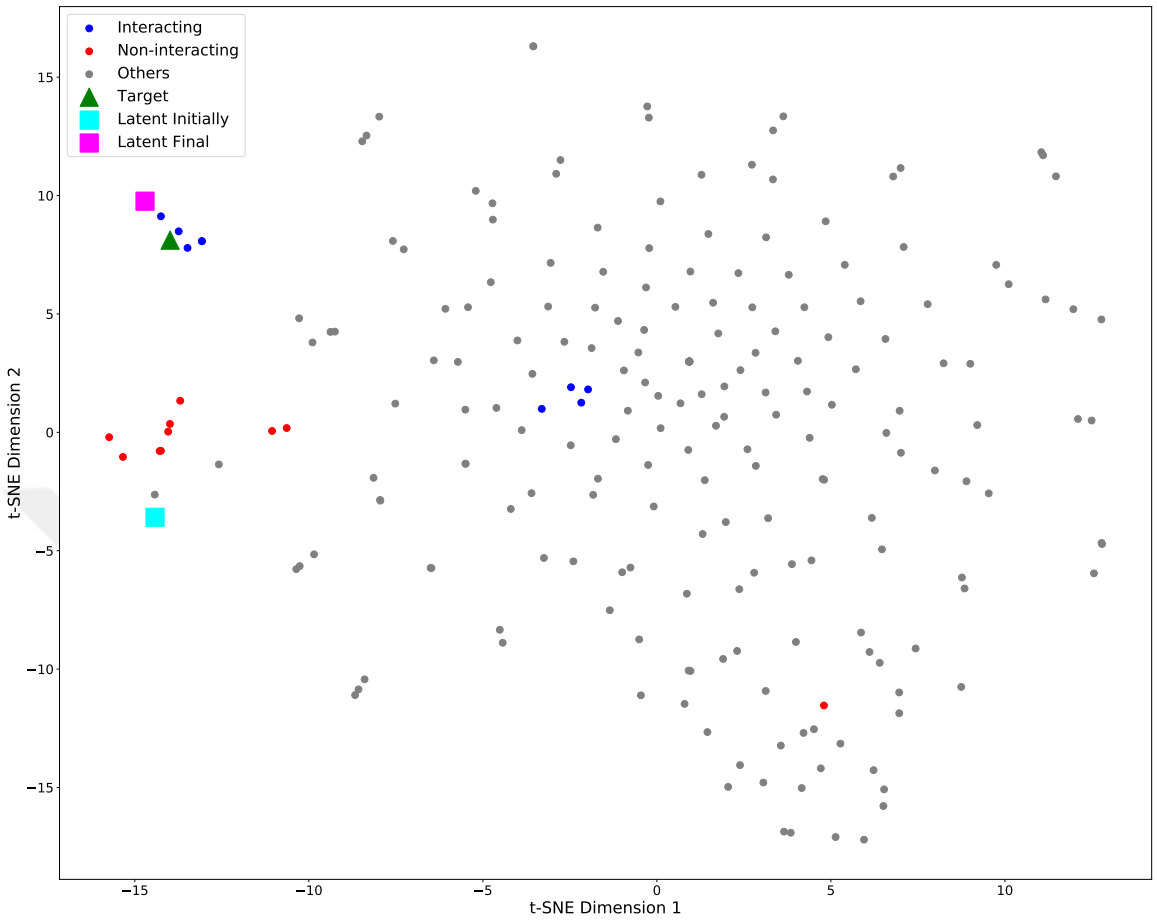

Figure C.3. Protein with UniProtID P49286.

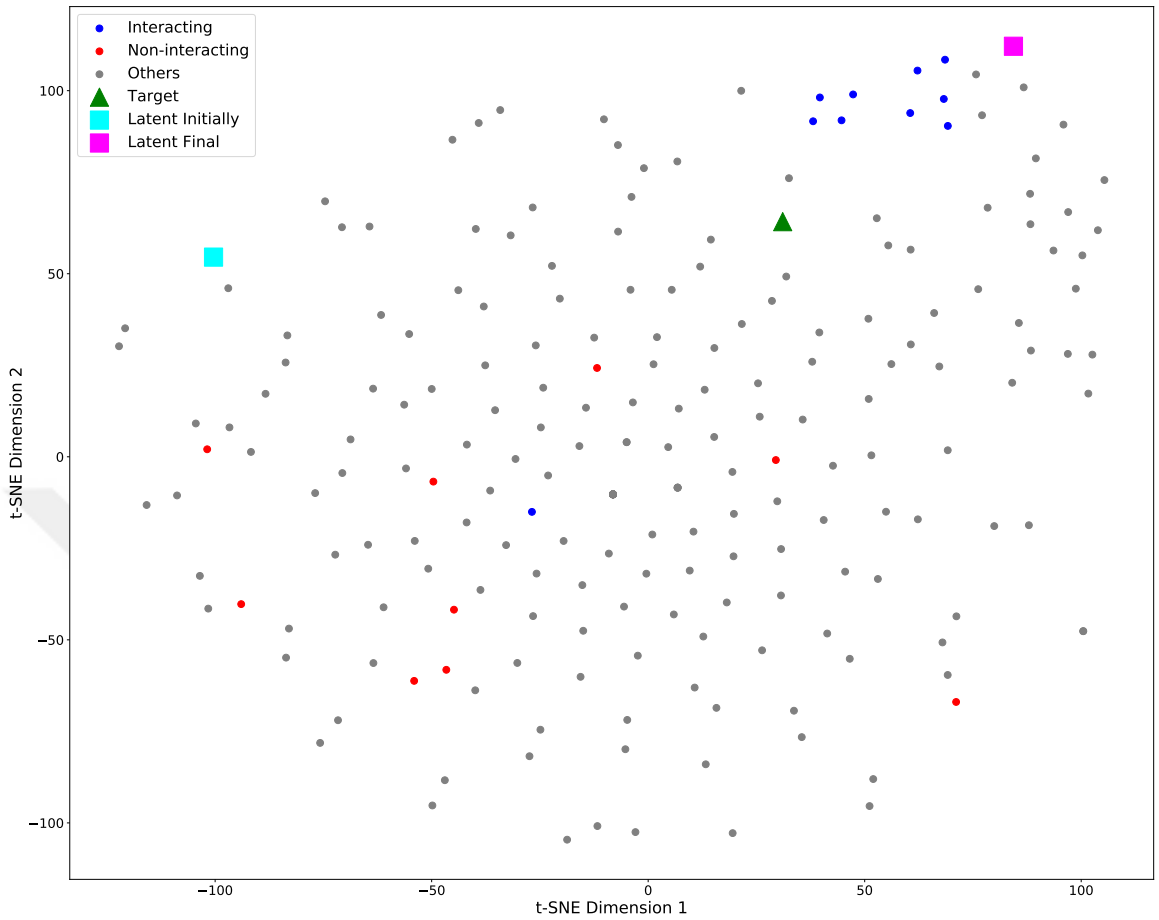

Figure C.4. Protein with UniProtID P05981.

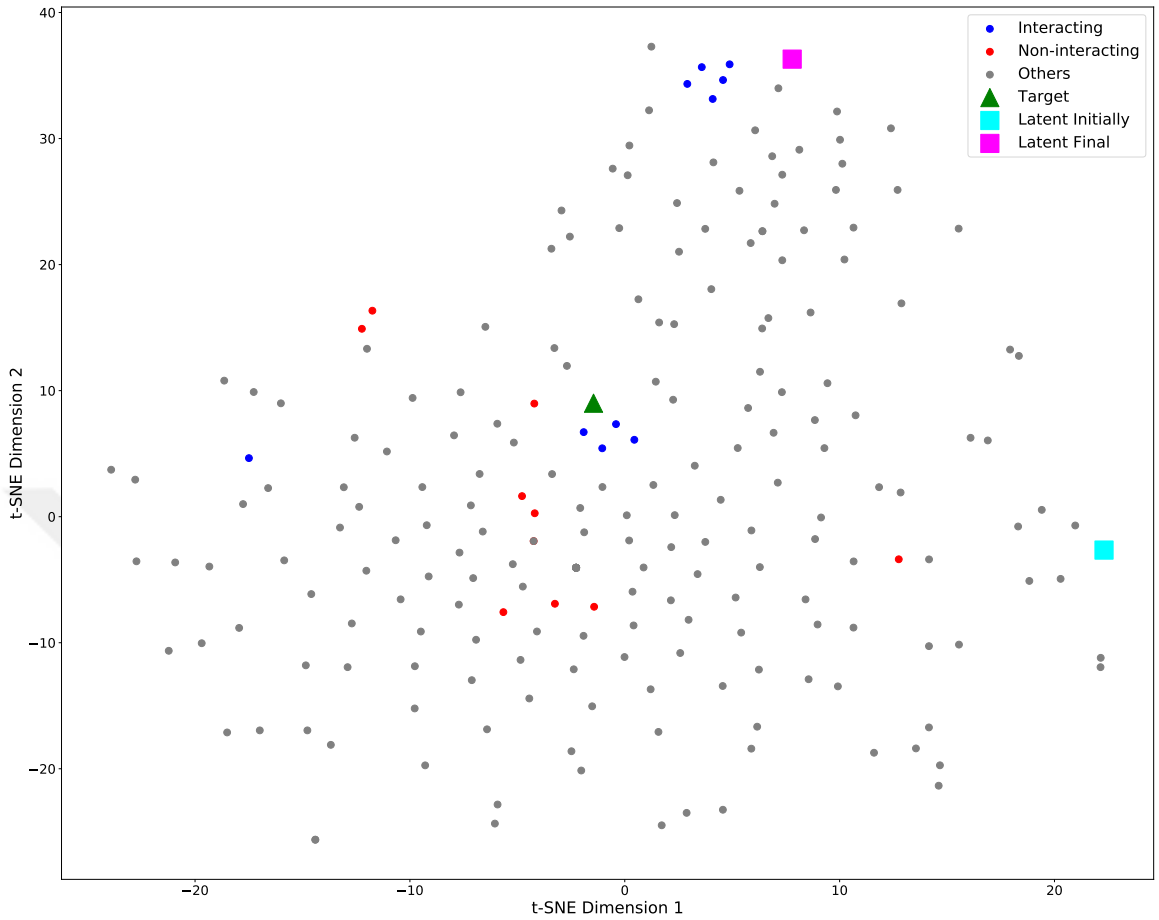

Figure C.5. Protein with P13612.

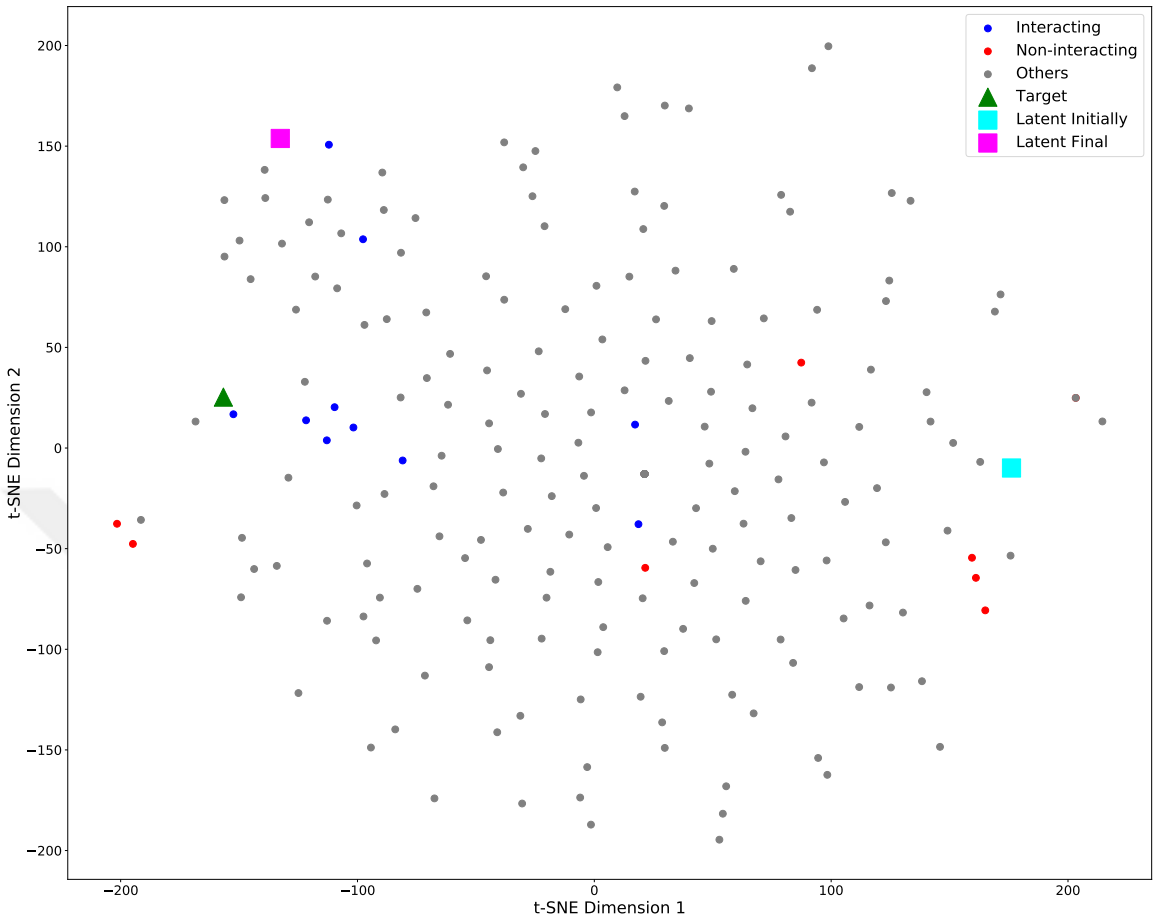

Figure C.6. Protein with Q16584.