SOLD: SELFIES-based Objective-driven Latent Diffusion *

Elbert Ho

The Pingry School

Abstract

Recently, machine learning has made a significant impact on *de novo* drug design. However, current approaches to creating novel molecules conditioned on a target protein typically rely on generating molecules directly in the 3D conformational space, which are often slow and overly complex. In this work, we propose SOLD (SELFIES-based Objective-driven Latent Diffusion), a novel latent diffusion model that generates molecules in a latent space derived from 1D SELFIES strings and conditioned on a target protein. In the process, we also train an innovative SELFIES transformer and propose a new way to balance losses when training multi-task machine learning models. Our model generates high-affinity molecules for the target protein in a simple and efficient way, while also leaving room for future improvements through the addition of more data.

1 Introduction

In the past few years, major advancements have been made in using machine learning for *de novo* drug design. Essentially, the goal is to generate novel molecules that are able to bind to a target protein with high affinity in order to inhibit its function, useful for developing new drugs. In the past, molecules were typically converted to one-dimensional (1D) SMILES strings and then fed into different models including LSTMs [6], VAEs [10], GANs [12], and GPTs [21]. SMILES strings were later replaced with SELFIES strings to solve the issue of generating invalid SMILES strings when the string does not follow the syntax of SMILES and cannot be turned into a molecule [19]. However, these 1D models are limited in that they do not capture all of the information in a molecule.

Recent work has shown the effectiveness of three-dimensional (3D) models in generating novel molecules. Initial models focused on autoregressive "growing" models that generated 3D structures bit by bit [9]. More recently, diffusion models have demonstrated superior performance in the domain of molecule generation. [16] [36] [32]. However, these models run in the extremely high dimensional atomic space which makes training and convergence difficult. To resolve this, Xu et al. [38] proposed a method to train a diffusion model in the latent space of a VAE. Still, this method is limited in that it is not designed to generate ligands with a specific target protein in mind, unlike other diffusion models that are conditioned on a target protein or binding pocket. Furthermore, though 3D models can incorporate more information, they rely on the use of complex equivariant graph neural networks (EGNNs) which increase model complexity and hurt convergence. In addition, this limitation means that they are unable to adapt many recent advancements in image generation models for use in drug discovery.

*This is a preprint of a paper to be submitted for publication.

In this work, we propose SELFIES-based Objective-driven Latent Diffusion, or SOLD, a novel 1D SELFIES latent diffusion model that is able to generate novel molecules conditioned on a target protein. We demonstrate that our model is able to generate novel ligands that are able to bind to a target protein with high affinity at a similar rate to other state-of-the-art approaches. Thus, although we may miss out on some 3D information, we find that the diffusion model is powerful enough to overcome this limitation. In other words, the model's 1D nature does not significantly impact its performance. To the best of our knowledge, this is the first work to successfully generate novel molecules using diffusion in the 1D space, and the first to generate molecules conditioned on a target protein using a latent diffusion model.

Our work is unique in that it is able to generate novel molecules simply and efficiently compared to other state-of-the-art (SOTA) models because of its 1D latent space. Additionally, our model is much more scalable to larger datasets as it does not require the 3D conformational information of the protein to be known due to a ligand-based-drug-discovery approach rather than a structure-based one.

2 Related Work

Conditional Diffusion Models. In the domain of image generation, recent works have focused on conditioning the model on either a target label or target text to generate realistic images that follow a prompt. In particular, the DALL-E models [29] [28] by OpenAI have been both popular and effective. However, our approach follows the GLIDE architecture proposed by Nichol and Dhariwal [26], as it is most easily adapted to the molecular domain.

Stable/Latent Generative Models. Due to the high complexity of the original space in many generative models, recent works have attempted to train models in a lower-dimensional latent space [17] [31]. These models are trained on a variety of domains including image generation and text generation. Most notably, Rombach et al. [31]. trained an image diffusion model in the latent space known as Stable Diffusion that has performed extremely well in image generation tasks. For molecule generation, Xu et al. [38] trained a diffusion model in the latent space of a VAE to generate novel molecules in the 3D space. Our study differs in that we generate molecules in the 1D space and condition on a target protein. We also do not use VAEs. We choose to use a latent space because past works on 1D molecular generation have shown that using the SMILES space directly is difficult and not very effective.

3 Background

3.1 SELFIES

SELFIES (Self-Referencing Embedded Strings) improve upon the SMILES (Simplified Molecular Input Line Entry System) representation of molecules [19]. In the original SMILES method, molecules are first represented as a graph where atoms are nodes and bonds are edges. The graph is then turned into a spanning tree by removing cycles such as benzene rings. Finally, the tree is traversed in a DFS manner to generate the SMILES string. However, this method is not perfect as there can be invalid SMILES strings. SELFIES strings, on the other hand, use formal grammar to ensure that all generated strings are valid. In particular, the parser is able to keep track of the potential valid next symbols and simply skip over the invalid ones. This makes SELFIES strings robust and more useful for generating novel molecules.

3.2 DDPMs

Denoising diffusion probabilistic models (DDPMs) were first introduced by Sohl-Dickstein et al. [34]. Diffusion algorithms model a probabilistic Markov chain that follows the distribution below:

$$q(z_t|z_{t-1}) := \mathcal{N}(\sqrt{\alpha_t}z_{t-1}, 1-\alpha_t\mathcal{I}) \tag{1}$$

A U-Net is used to model the reverse diffusion process represented by

$$p_\theta(z_{t-1}|z_t, y) := \mathcal{N}(\mu_\theta(z,t,y), \Sigma_\theta(z,t,y)) \tag{2}$$

where y is the label or condition. The trained backward model is then used to generate samples by starting from $z_t \sim \mathcal{N}(0,\mathcal{I})$ and using $p_\theta(z_{t-1}|z_t,y)$ to go to z_{t-1} and then $p_\theta(z_{t-2}|z_{t-1},y)$ to go to z_{t-2} and so on until we reach z_0 which is our final sample.

In practice, the model is optimized according to the improved loss function proposed by Nichol and Dhariwal [25] below:

$$L_{vlb} := L_0 + L_1 + \cdots + L_T \tag{3}$$
$$L_0 := -\log p_\theta(z_0|y) \tag{4}$$
$$L_{t-1} := D_{KL}(q(z_{t-1}|z_0)||p_\theta(z_{t-1}|z_t,y)) \tag{5}$$
$$L_T := D_{KL}(q(z_T|z_0)||p(z_T)) \tag{6}$$
$$L_{simple} := \mathbb{E}_{t,z_0,\epsilon}[||\epsilon - \epsilon_\theta(z_t,t,y)||^2] \tag{7}$$
$$L_{final} := L_{simple} + \lambda L_{vlb} \tag{8}$$

Essentially, the simple loss is the mean squared error between the predicted noise and the actual noises. The VLB loss, on the other hand, is the sum of the KL divergences between the predicted distribution and the true distribution at each time step. L_0 is defined as the negative log likelihood of the distribution. It is used to help the model learn the standard deviation of the distribution rather than just the mean. λ is a hyperparameter that controls the weight of the VLB loss, and it is set to a low value (.0001) to prevent it from dominating the loss. When sampling, μ_θ is derived from ϵ_θ [13].

3.3 Classifier-Free Guidance

Classifier-free guidance is a simple method proposed by Ho & Salimans [14] to improve the performance of conditional diffusion models. The idea is to train the model to recognize the general distribution of all possible samples by occasionally replacing y with $\emptyset$ when training the model. When sampling, we use

$$\tilde{\epsilon}_\theta(z,t,y) = (1+w)\epsilon_\theta(z,t,y) - w\epsilon_\theta(z,t,\emptyset) \tag{9}$$

so that the final sample is closer to the conditioned distribution than to the unconditional distribution. Increasing w increases the guidance.

4 Methods

Inspired by stable diffusion models, we train and use a latent diffusion model to generate novel molecules. Figure 1 is a graphical pipeline of the model, showing how molecules are encoded, decoded, and then generated. We first represent molecules as SELFIES strings and use a transformer to learn a latent representation of the molecule in Section 4.1. We then detail the specifics of the diffusion model in Section 4.2.

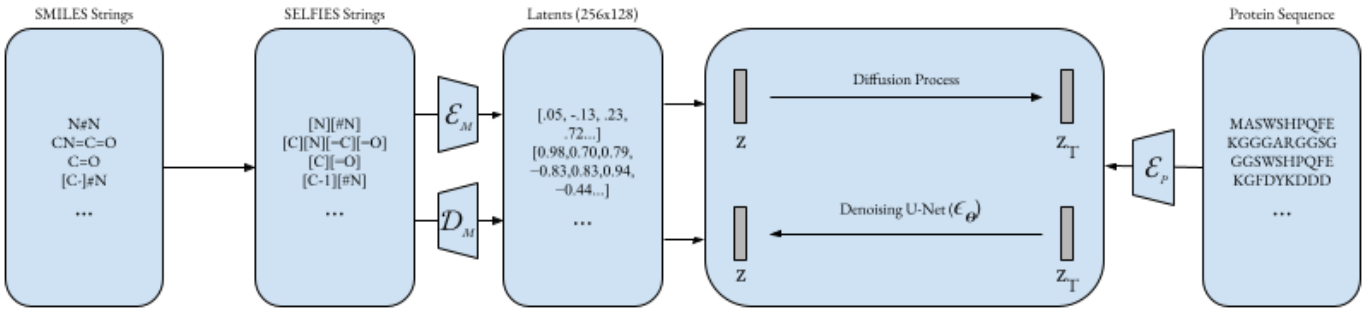


Figure 1: Illustration of SOLD. Molecules are first encoded as SELFIES strings and then transformed into the latent space through a transformer. A diffusion model is then trained in the latent space to generate novel molecules. Proteins are encoded as ESM-2 embeddings on the right.

4.1 Molecular Encoder

Model Architecture. Since transformers [35] have been shown to perform extremely well on NLP tasks [4] and have also been shown to be effective when applied to SMILES strings [15] [8], we use a transformer to learn representations of SELFIES strings. We have attempted to use a VAE instead to exploit the inherently probabilistic nature of its latent space but found that the VAE was unable to learn a sufficiently accurate enough latent representation (our experiments showed that 0% of sequences were correctly reconstructed). In particular, while VAEs may be able to learn an approximation of a latent space, they are fundamentally flawed for our purposes as they are not able to reconstruct inputs very well due to their stochastic nature. Thus, our VAE begins to act as a generator rather than just an encoder and decoder, which could interfere with our diffusion model.

In comparison, transformers can learn an effective reconstruction of the latent space because their attention mechanism allows them to learn the relationships between different parts of the SELFIES string and thus generate a more accurate latent representation. Our model has 2 transformer layers each with embedding dimension 256 and 4-heads of attention.

Training. We sourced 10,000 molecules from the CHEMBL dataset [40]. First, molecules were converted to SMILES and then randomized to avoid biasing the model towards the canonical SMILES sourced from CHEMBL [1]. Then, molecules were converted to SELFIES strings because SELFIES strings are guaranteed to be valid, something that is particularly useful when generating new molecules [19]. SELFIES strings were first tokenized using the standard SELFIES tokenizer. We then trained and used a byte pair encoder (BPE) [33] as we needed to trade-off latent space size for vocabulary size. This is because we will be training the diffusion model on the latent space and we would rather use a smaller latent space later on. In particular, we tried to train without byte pair encoding but this forced us to either truncate the SELFIES strings (which would be problematic as the generative model would have to understand incomplete molecules) or use a much larger latent space to fit all of the SELFIES strings. Our BPE model had a vocabulary size of 256 as we found this to reduce most of the SELFIES strings to a length less than 128. All SELFIES strings were then tokenized using the SELFIES tokenizer and BPE tokenizer before being padded to length 128 or simply removed from the dataset if they had a length longer than 128.

As suggested by Honda et al. [15], we train the model in a multi-task setting where the final layer of the transformer is used to predict the SPS (Spacial Score), MinEStateIndex (electrotopological state index), ExactMolWt (exact molecular weight), BalabanJ (Balaban J index), and VSA_EState6 (EState fragment contribution to the sixth bin of the VSA_EState). These tasks were chosen using PCA analysis on the available RDKit [20] molecular descriptors.

In essence, given an encoder $\mathcal{E}$ that maps SELFIES tokens to a latent representation and a decoder $\mathcal{D}$ that maps latent representations to SELFIES tokens, we train the model to minimize the standard cross-

entropy reconstruction loss $\ell(\mathcal{D}(\mathcal{E}(x)), x)$ as well as task-specific losses $\ell(\mathcal{D}_t(\mathcal{E}(x)), y_t)$ where y_t is the target value for task t (all mean squared error).

To optimize, we modify the multi-task loss training approach proposed by Lin et al. [22], which uses a momentum-based approach combined with gradient normalization, by integrating the second moment found in the Adam optimizer [18]. We found that adding the second moment significantly improved convergence.

We outline our proposed novel training function below in Algorithm 1, where the second moment is represented by ν in lines 8 and 16.

Algorithm 1 Dual-Balancing Multi-Task Learning with Adam Modifications

Require: numbers of iterations T, learning rate η, tasks $\{\mathcal{D}_k\}_{k=1}^K$, β, β_2, $\epsilon = 10^{-8}$

1: randomly initialize θ_0, $\{\psi_{k,0}\}_{k=1}^K$

2: initialize $\hat{g}_{t,-1} = 0$, for all t

3: **for** $t = 0, \ldots, T-1$ **do**

4: **for** $k = 1, \ldots, K$ **do**

5: sample a mini-batch dataset $\mathcal{B}_{k,t}$ from $\mathcal{D}_k$

6: $g_{k,t} = \nabla_{\theta_t} \log(\ell_k(\mathcal{B}_{k,t}; \theta_t, \psi_{k,t}) + \epsilon)$

7: compute $\mu_{k,t} = \frac{\beta\mu_{k,t-1} + (1-\beta) g_{k,t}}{1-\beta^t}$

8: compute $\nu_{k,t} = \frac{\beta_2\nu_{k,t-1} + (1-\beta_2) g_{k,t}^2}{1-\beta_2^t}$

9: $\hat{g}_{k,t} = \frac{\mu_{k,t}}{\sqrt{\nu_{k,t}} + \epsilon}$

10: **end for**

11: compute $\tilde{g}_t = \alpha_t \sum_{k=1}^K \frac{\hat{g}_{k,t}}{\|\hat{g}_{k,t}\|_2 + \epsilon}$, where $\alpha_t = \max_{1 \le k \le K} \|\hat{g}_{k,t}\|_2$

12: update task-sharing parameter by $\theta_{k+1} = \theta_k - \eta \tilde{g}_k$

13: **for** $k = 1, \ldots, K$ **do**

14: $g'_{k,t} = \nabla_{\theta_t} \log(\ell_k(\mathcal{B}_{k,t}; \theta_t, \psi_{k,t}) + \epsilon)$

15: compute $\mu'_{k,t} = \frac{\beta\mu'_{k,t-1} + (1-\beta) g'_{k,t}}{1-\beta^t}$

16: compute $\nu'_{k,t} = \frac{\beta_2\nu'_{k,t-1} + (1-\beta_2) g'^2_{k,t}}{1-\beta_2^t}$

17: $\psi_{k,t+1} = \psi_{k,t} - \eta \frac{\mu'_{k,t}}{\sqrt{\nu'_{k,t}} + \epsilon}$

18: **end for**

19: **end forreturn** θ_T, $\{\psi_{k,T}\}_{k=1}^K$

The model was trained for a total of 200 epochs with 20 epochs used for pretraining without the task losses. This was done to help the model first learn a robust latent representation before simultaneously learning the tasks. Cosine annealing was used for the learning rate which was set to start at .0001.

4.2 Latent Diffusion Model

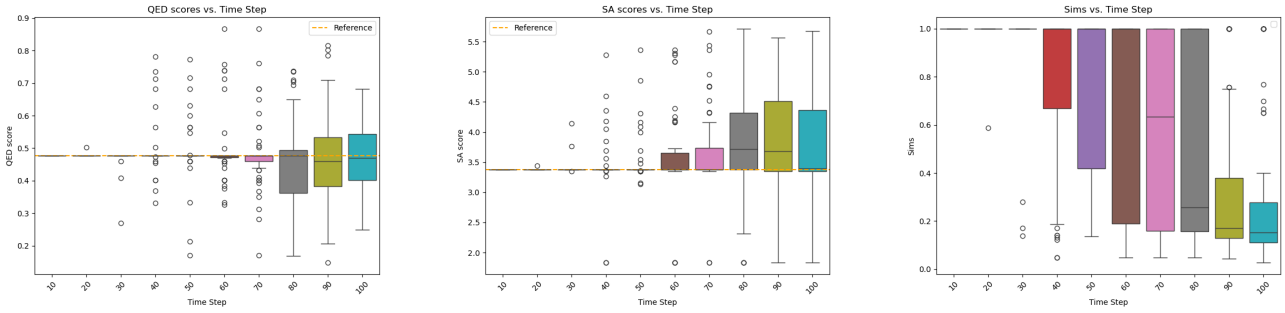
Model Architecture. Once the SELFIES strings are encoded into the latent space, we train a diffusion model to generate novel molecules. We take the last layer of the transformer as our encoder $\mathcal{E}_M$ which makes the latents essentially length 128 1D vectors that have 256 channels. Before inputting into the model, we

normalize the latents to have a mean of 0 and a standard deviation of 1 (using a global mean and standard deviation) before converting to 0 to 1 using the standard normal cumulative distribution function. In other words, we calculate $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{t^2}{2}} dt$ which outputs in the range 0 to 1. Finally, the input is linearly scaled to the range of -1 to 1. When going back to our original space, we reverse the process by applying the inverse normal cumulative distribution function and truncate the final values to be between -6.5 and 6.5 because of precision. We imitate the architecture proposed by Nichol and Dhariwal [26] with the exception that we use a 1D U-Net instead of a 2D one. We used 1000 timesteps for the diffusion process. In addition, instead of using text embeddings from a transformer as the conditioning for the model, we use ESM-2 [23] with embedding dimension of 1280 to embed proteins. ESM-2 is a transformer-based model that can be used to turn the sequence of amino acids into an embedding. The ESM-2 model's weights are trained during the diffusion process.

Training. Our model has 360M parameters and was trained on a dataset of 15,000 protein-ligand pairs sourced from the PDBbind dataset [37]. The model was trained for a total of 1000 epochs with batch size of 32 with 16-bit precision on an NVIDIA RTX 4090. Additionally, we trained with classifier-free guidance so 20% of inputs were conditioned on $\emptyset$.

Sampling. To sample from the model, we start with a random latent vector with dimensions of 256 x 128. We then use the reverse diffusion model combined with classifier-free guidance to generate a sample. We found experimentally that a guidance weight of 5 worked the best. Latent samples were converted back to SELFIES strings through the decoder $\mathcal{D}_M$ and then converted to SMILES strings which could then be converted to either 2D images or 3D structures via RDKit.

Optimizing Qualities. To optimize the quality of generated molecules, we attempted to adopt the simple evolutionary algorithm proposed by Schneuing et al. [32]. In particular, given a molecule with a desired binding affinity, we noise the latent representation by 75 steps before denoising it back to a molecule. In this way, we obtain similar molecules with slightly different properties. By selecting the best molecules with desired properties (such as QED and SAS), we can once again noise these molecules and then denoise them to obtain even better molecules. This process is repeated until the desired properties are met. We chose to noise by 75 steps because we found that this was the threshold where the molecules would be similar to the original but also have some diversity (Figure 2). Note that in Figure 2b, we use raw SA scores so lower is better and the range is 1 to 9. Here, rather than using $w = 5$ for the guidance weight as before, we lower it to $w = 0$, as increasing w was found to lower diversity which is not desirable in this case. Though, as discussed later, we found that the evolutionary algorithm did not significantly improve the quality of the molecules and actually hurt the Vina score considerably.



(a) QED Scores when noising and denoising from 0 to 100 steps

(b) SAS Scores when noising and denoising from 0 to 100 steps

(c) Similarity to original molecule when noising and denoising from 0 to 100 steps

Figure 2: Some results of the evolutionary algorithm.

5 Experiments

Baselines. We compare our model to the following SOTA baselines. **3D-SBDD** [24] was one of the first 3D conditional molecule generative models, and it uses an autoregressive approach to sample the 3D coordinates of each atom in a molecule from a predicted probability distribution. **Pocket2Mol** [27] is a similar approach except it takes into consideration bond types and functional groups. **TargetDiff** [11] and **DiffSBDD-cond** [32] are similar 3D diffusion models that are conditioned on a target protein pocket. There were also a few models that we looked at such as DrugGPT which is a relatively recent 1D model that uses a GPT-like architecture rather than diffusion to generate molecules. However, since none of their papers reported Vina score as a metric and we were unable to run their model with our limited resources, we do not include them in the comparisons below.

Metrics. We evaluate our model on the following metrics.

- **Vina** is the binding energy of the generated ligand with the target protein or protein-pocket as predicted by AutoDock Vina. We used UniDock [39] (a GPU accelerated version of AutoDock) for our experiments, but the results are comparable. Additionally, we used the standard parameters (exhaustiveness: 128, max step: 20, num modes: 3, refine step: 3, top n: 100). Due to compute constraints, we only evaluate our model on a few select proteins not in the training set and take the average. In particular, we show results for our tests with the COVID-19 3C-like main protease ($3CL^{\text{pro}}$), which is a known target for COVID-19 drugs [41].
- **QED** [3] (quantitative estimate of drug-likeness) is a metric that measures how "drug-like" a molecule is based on a weighted sum of different molecular properties such as logP and molecular weight.
- **SA** [7] (synthetic accessibility score) calculates how "easy" it would be to synthesize the molecule based on composite fragments. Here, scores are normalized between 0 and 1 and reversed so that higher scores are better.
- **Diversity** calculate the difference between generated molecules using the Tanimoto similarity between Morgan fingerprints [2].
- **Time** is the time in seconds taken to generate 100 molecules for a given protein or protein-pocket. Since this time is dependent on the GPU, we scale the time based on the GPU used. In particular, the V100 is close in speed to the RTX 3060 that we used for inference, but the A100 used for DiffSBDD is about 8 times faster. Thus, we scale their time up by 8.

w	# Filter (↑)	Vina (↓)	Vina (Top-10%) (↓)	Filtered (↓)	Filtered (Top-10%) (↓)
0	16	-4.103	-6.771	-4.282	-5.957
1	9	-4.416	-7.709	-5.281	-8.197
3	7	-4.486	-7.650	-5.193	-6.731
5	13	-4.498	-9.830	-4.900	-6.613
7	12	-4.558	-8.670	-4.257	-5.982
9	11	-3.415	-6.600	-5.133	-6.875
11	12	-2.935	-6.429	-4.609	-7.131

Table 1: Summary of Results for Different Values of w.

First, we evaluate the effect of the guidance weight w on the performance of our model (Table 1). We do this by testing w from 0 to 11 and evaluating the Vina score. We report two different metrics: the first is the entire set of generated molecules, and the second is a "filtered" list where we only take molecules of QED at least 0.40 and SAS at least 0.33. We find that $w = 5$ performs the best in terms of Vina score and top 10% for the overall set of molecules. Note that the filtered metrics have large uncertainty due to small sample size, so we choose to rely more on the overall Vina metrics. Therefore, we use $w = 5$ for all remaining experiments.

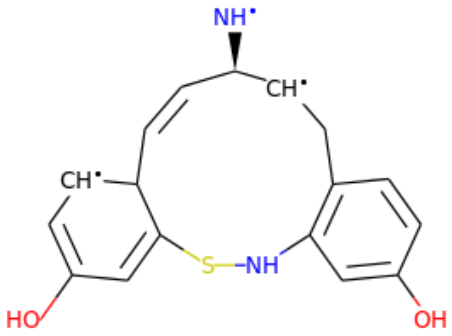
Model	**Vina (↓)**	**Vina (Top-10%) (↓)**	**QED (↑)**	**SA (↑)**	**Div. (↑)**	**Time (↓)**
3D-SBDD [24]	-5.888	-7.289	0.502	0.675	0.742	19659
Pocket2Mol [27]	-7.058	-8.712	**0.572**	**0.752**	0.735	2504
TargetDiff [11]	-7.318	-9.669	0.483	0.584	0.718	3428
DiffSBBDD [32]	**-7.333**	**-9.927**	0.475	0.612	0.725	1088
SOLD (Ours)	-4.498	-9.830	0.3727	0.451	**0.946**	**960**

Table 2: Comparison of SOTA models for molecular generation tasks.

We find that our model performs comparably to other existing models (Table 2). In particular, though our average Vina score is slightly lower than other models, our top 10% Vina score is comparable. Since drug discovery primarily depends on the best hits, we believe that the weakness in the average Vina score is not a major issue. Additionally, our model is able to generate molecules faster than other SOTA models. Combined with our higher diversity, we are able to therefore quickly explore a much larger chemical space. As such, we believe that using SMILES strings as an input can be advantageous because it is much easier to explore a larger chemical space with 1D models than 3D models.

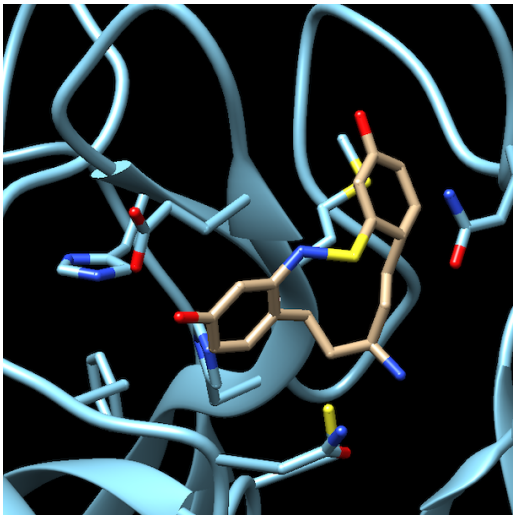
Below, we show two generated molecules for the COVID-19 $3CL^{\text{pro}}$. In Figures 3 and 4, we show the 2D rendering of the generated molecule as well as its location in the larger protein and in the binding pocket. We also show the hydrophobic surface of the binding pocket with the bound ligand. For comparison, Figure 5 shows the known drug Nirmatrelvir, which is a part of the Paxlovid drug combination [30]. As can be seen, our model is able to generate molecules that have similar QED and SAS scores to known drugs but much better docking scores.

In particular, our docking score is intended to be a relative estimate of ΔG. However, since Autodock Vina does not include the significant entropy increase from the ejection of water molecules from the binding site, it will be off by a constant factor. Here, the experimental ΔG of Nirmatrelvir is about -11.5 [5] so we also subtract 5.60 from the ΔG of our compound (since our Vina output for Nirmatrelvir was -5.90). We have that IC50 (half-maximal inhibitory concentration) is approximately equivalent to K_i and we can relate K_i to the true ΔG of the binding. We have that $IC_{50} \approx K_i = e^{\frac{\Delta G}{RT}}$ since $K_i = \frac{1}{K_d}$ and $K_d = e^{-\frac{\Delta G}{RT}}$ where R is the gas constant and T is the temperature. Calculating this value out for the known drug Paxlovid ($\Delta G \approx -11.5$) and our first molecule ($\Delta G \approx -13.4$), we get that the IC_{50} of Paxlovid is approximately 7.8 nM and our first molecule is approximately 0.37nM. Thus, our model is able to generate molecules that are more than 20 times more potent than already effective drugs.

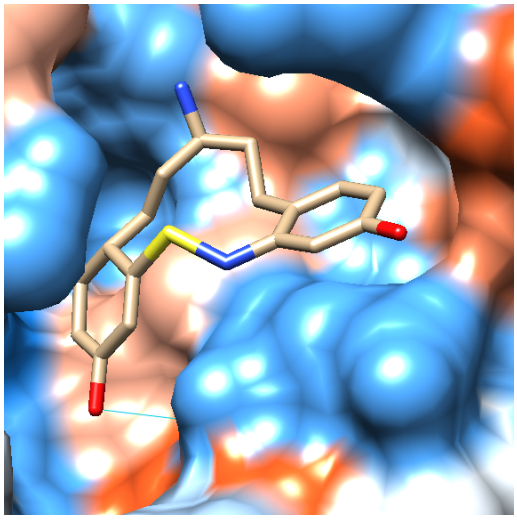


(a) Illustration of one generated molecule. This molecule had a QED of 0.505 and an SAS of 0.374 and an affinity score of -7.70.

(b) Rendering of molecule bound to $3CL^{\text{pro}}$ protease

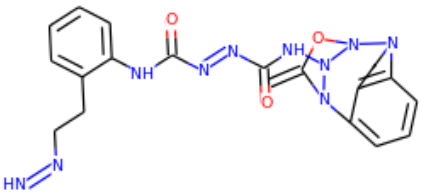


(c) Rendering of molecule with $3CL^{\text{pro}}$ protease binding site

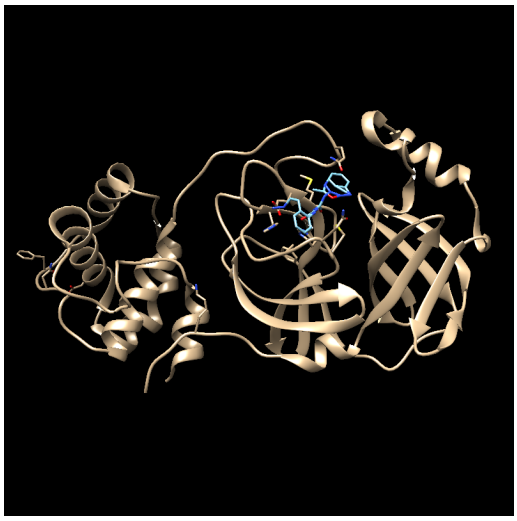


(d) Hydrophobic surface view of binding site with bound ligand

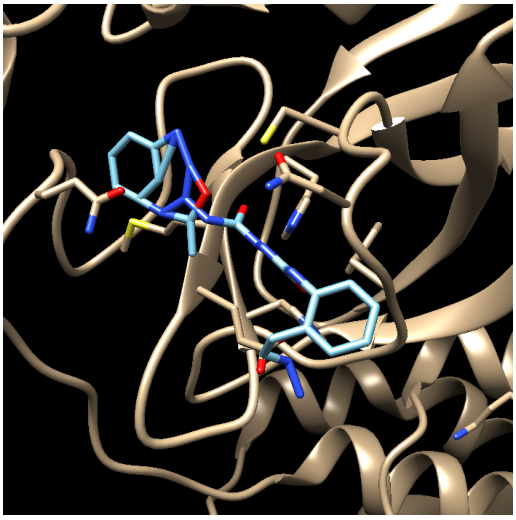
Figure 3: Renderings of molecule generated by SOLD



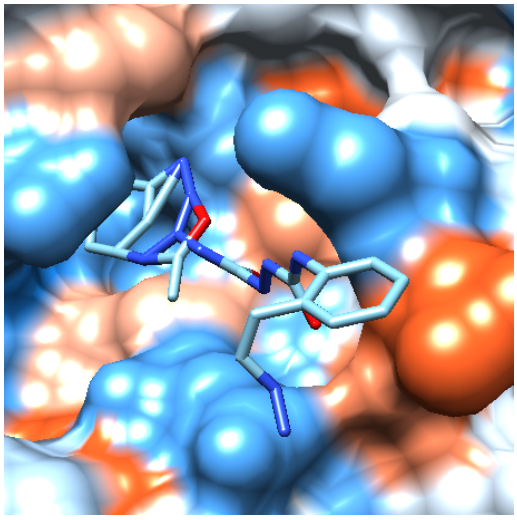
(a) Illustration of one generated molecule. This molecule had a QED of 0.594 and an SAS of 0.363 and an affinity score of -7.15.



(b) Rendering of molecule bound to $3CL^{\text{pro}}$ protease

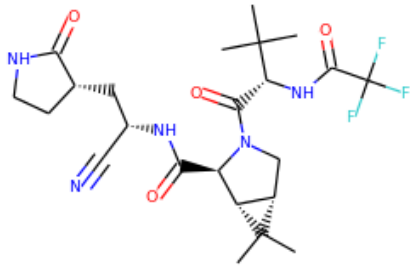


(c) Rendering of molecule with $3CL^{\text{pro}}$ protease binding site

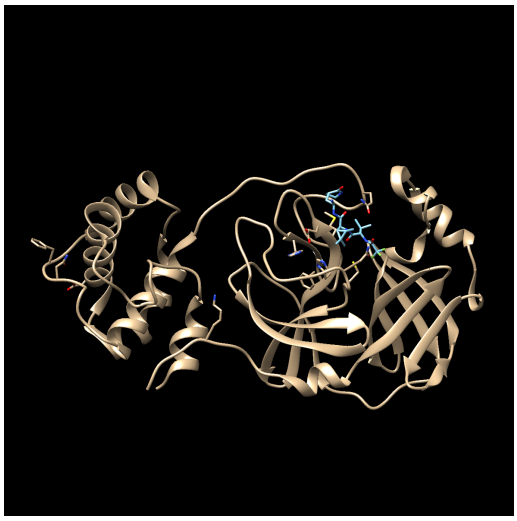


(d) Hydrophobic surface view of binding site with bound ligand

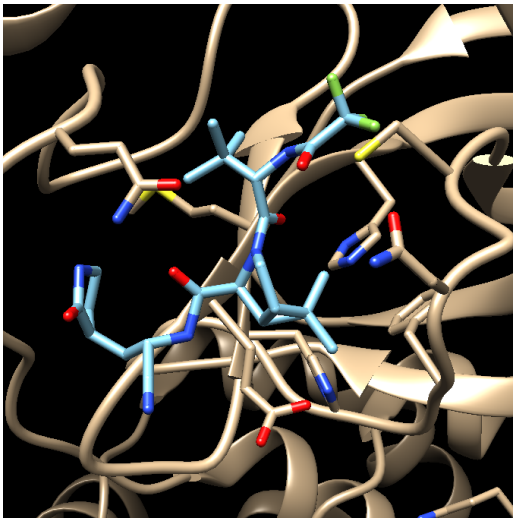
Figure 4: Render of another molecule generated by SOLD



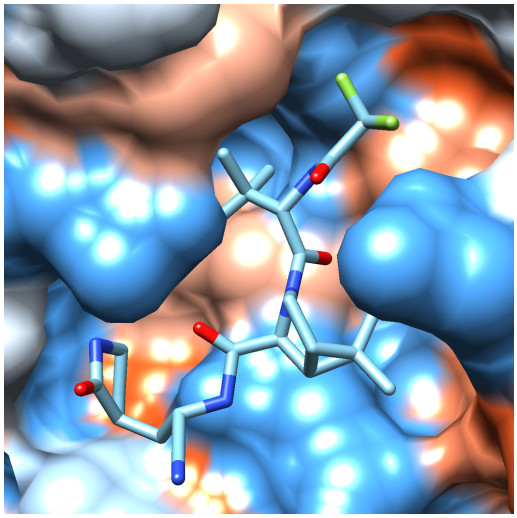
(a) Illustration of Paxlovid (Nirmatrelvir). This molecule had a QED of 0.504 and an SAS of 0.491 and an affinity score of -5.90.



(b) Rendering of molecule bound to $3CL^{\text{pro}}$ protease



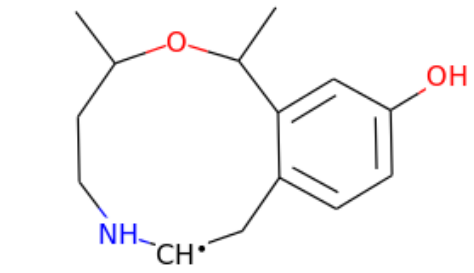
(c) Rendering of molecule with $3CL^{\text{pro}}$ protease binding site



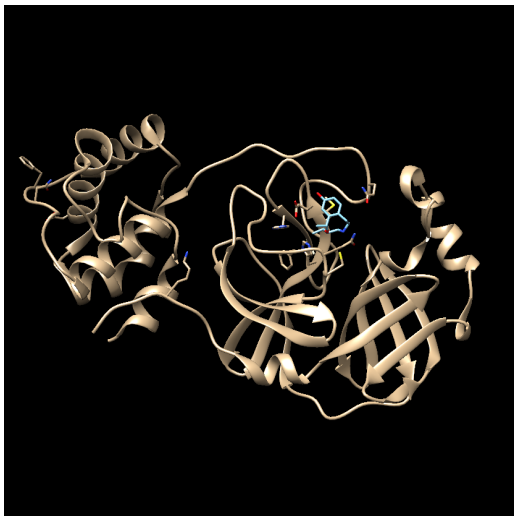
(d) Hydrophobic surface view of binding site with bound ligand

Figure 5: Renderings of Paxlovid

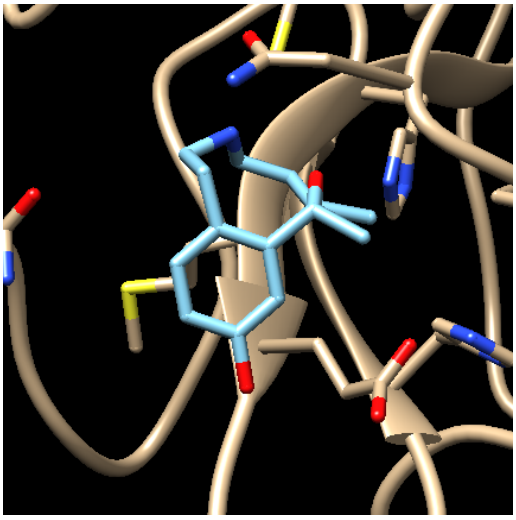
Property Optimization. Using $3CL^{\text{pro}}$, we also evaluated the effectiveness of the property optimization algorithm proposed by Schneuing et al. [32] as described above. To do this, we first obtain a sample with high binding affinity but low QED and high SAS. We find that this method of optimization is not very effective in our case. In particular, while we are able to increase our QED and SAS scores after one generation, we find that docking scores drop off significantly. In addition, after 5 generations, the QED and SAS have not improved significantly and the molecules are much more dissimilar to the original, leading to even worse docking scores. For example, the molecule in Figure 6a is derived 3a after one generation. It has a QED of 0.725, an SAS of 0.493, and an affinity score of -5.65. After 5 generations, we attain the molecule in Figure 7a with a QED of 0.644, an SAS of 0.395, and an affinity score of -5.23. As a result, we did not adopt property optimization in our final approach.



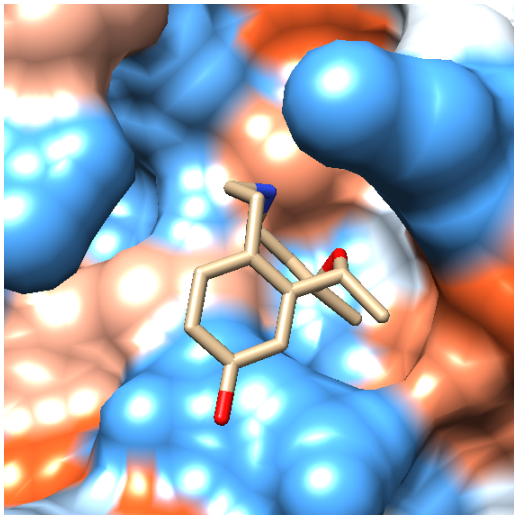
(a) Illustration of molecule. This molecule had a QED of 0.725 and an SAS of 0.493 and an affinity score of -5.65.



(b) Rendering of molecule bound to $3CL^{\text{pro}}$ protease

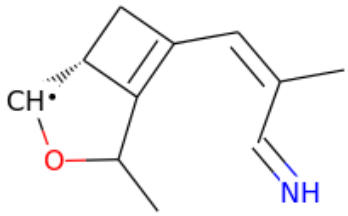


(c) Rendering of molecule with $3CL^{\text{pro}}$ protease binding site

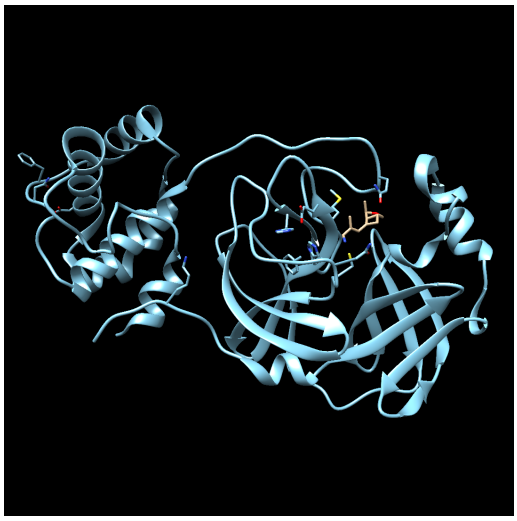


(d) Hydrophobic surface view of binding site with bound ligand

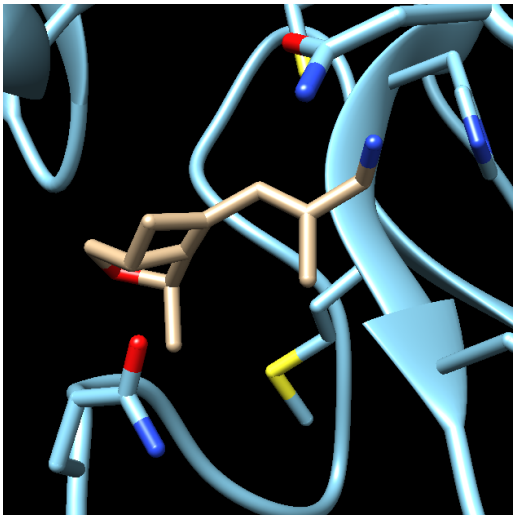
Figure 6: Renderings for molecule after 1 generation



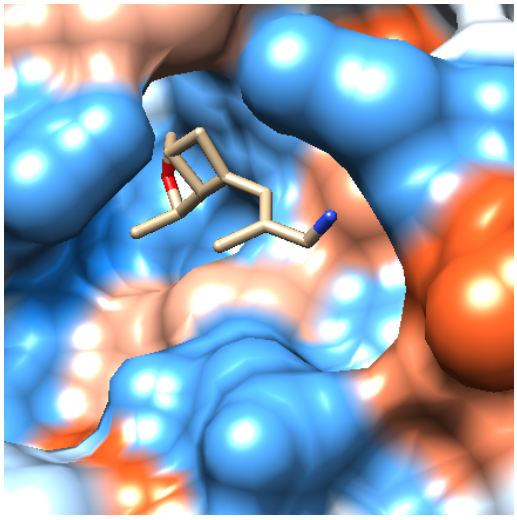
(a) Illustration of molecule. This molecule had a QED of 0.644 and an SAS of 0.395 and an affinity score of -5.23



(b) Rendering of molecule bound to $3CL^{\text{pro}}$ protease



(c) Rendering of molecule with $3CL^{\text{pro}}$ protease binding site



(d) Hydrophobic surface view of binding site with bound ligand

Figure 7: Renderings for molecule generated after 5 generations

6 Conclusion and Future Work

In this paper, we propose a novel method for *de novo* drug discovery using a latent diffusion model in the 1D SMILES/SELFIES space and demonstrated its potential to generate molecules with potency comparable to or exceeding known drugs. Our model stands out from existing approaches for the following reasons:

- We use a novel transformer model to learn a latent representation of the 1D SMILES/SELFIES space (previous models have never used an encoder-decoder transformer trained in a multi-task fashion for molecular generation).
- We perform drug discovery with diffusion using 1D SMILES/SELFIES (previous diffusion models were

either in 2D or 3D).

- We use latent diffusion models for target-aware de novo drug discovery (previous models either did not use latent diffusion or did target-agnostic drug discovery if latent diffusion was used).

Compared to other state-of-the-art models, our approach has the following advantages:

- The model is able to generate molecules significantly faster than other SOTA models. This is because the model is using a much simpler 1D representation and does not require the generation of 3D coordinates. The time to generate can also be halved if accuracy is less of a concern and we perform inference without classifier-free guidance.
- The model has the highest diversity of any SOTA model. This is because the 1D latent nature of the model allows it to explore a larger space. By exploring more molecules, we can find better hits, which is also why we have a relatively high Vina Top-10% score.
- As we only require the amino acid sequence of the target protein, which is much easier to obtain than an accurate 3D structure, we are not only able to train on a much larger dataset if given more computing resources, but we can also perform inference on a much larger set of proteins.

There are also a few limitations with our approach:

- The model is not able to optimize properties using the method proposed by Schneuing et al. [32] very well. Unfortunately, the SAS and QED are somewhat random and the model quickly loses its affinity score, even with only small changes in the latent space.
- The model has a lower average Vina score than other SOTA models. This means that we must generate more molecules to find a good hit. However, since our model is fast and has the highest Vina Top-10%, we believe that this might not be a major issue.
- The model has relatively low average QED and SAS. This is likely connected to the fact that the model explores a very large chemical space. Still, this may not be a major issue since many real drugs have low QED and SAS scores and, in addition, small manual tweaks can be made to the generated molecules to increase these scores.

Future work could improve our approach in the following ways:

- Further optimize the model by increasing the amount of data in the training. We found that the model was able to generate better molecules as we increased the training dataset size, but we were limited by our resources to only train on 15,000 protein-ligand pairs. Expanding to a larger portion of the PDBBind dataset could potentially improve performance.
- Increase the size of the diffusion model and ESM encoding. We found that the model was able to generate better molecules with more parameters, but were limited by our resources.
- Develop a better protein targeting approach to address the suboptimal average Vina score, enabling a more efficient search process without relying solely on a large search space.
- Evaluate the toxicity of generated compounds by looking at potential off-target interactions. This would be another step in making our molecules as realistic as possible.

7 Acknowledgments

This research was conducted entirely independently by the author. We thank the Pingry School for generously providing us with the computing resources that enabled us to run our experiments.

References

[1] Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminf.*, 11, 2019.

[2] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminf.*, 7(1):1–13, December 2015.

[3] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nat. Chem.*, 4(2):90–98, January 2012.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[5] Andrea Citarella, Alessandro Dimasi, Davide Moi, Daniele Passarella, Angela Scala, Anna Piperno, and Nicola Micale. Recent advances in sars-cov-2 main protease inhibitors: From nirmatrelvir to future perspectives. *Biomolecules*, 13(9), 2023.

[6] Peter Ertl, Richard Lewis, Eric J. Martin, and Valery R. Polyakov. In silico generation of novel, drug-like chemical matter using the LSTM neural network. *CoRR*, abs/1712.07449, 2017.

[7] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, Jun 2009.

[8] Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks, 2020.

[9] Niklas W. A. Gebauer, Michael Gastegger, and Kristof T. Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules, 2020.

[10] Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016.

[11] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

[12] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models, 2018.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[15] Shion Honda, Shoi Shi, and Hiroki R. Ueda. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery, 2019.

[16] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d, 2022.

[17] Sanket Kalwar, Animikh Aich, and Tanay Dixit. Latentgan autoencoder: Learning disentangled latent distribution, 2022.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[19] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry. *CoRR*, abs/1905.13741, 2019.

[20] Greg Landrum. Rdkit: Open-source cheminformatics, 2006–. Accessed: 2024-09-07.

[21] Yuesen Li, Chengyi Gao, Xin Song, Xiangyu Wang, Yungang Xu, and Suxia Han. Druggpt: A gpt-based strategy for designing potential ligands targeting specific proteins. *bioRxiv*, 2023.

[22] Baijiong Lin, Weisen Jiang, Feiyang Ye, Yu Zhang, Pengguang Chen, Ying-Cong Chen, Shu Liu, and James T. Kwok. Dual-balancing for multi-task learning, 2023.

[23] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2022.

[24] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design, 2022.

[25] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

[26] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022.

[27] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets, 2022.

[28] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[29] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[30] J Reina and C Iglesias. Nirmatrelvir plus ritonavir (paxlovid) a potent SARS-CoV-2 3CLpro protease inhibitor combination. *Rev. Esp. Quimioter.*, 35(3):236–240, June 2022.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[32] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models, 2023.

[33] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016.

[34] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[36] Clement Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation, 2023.

[37] Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The PDBbind database: methodologies and updates. *J. Med. Chem.*, 48(12):4111–4119, June 2005.

[38] Minkai Xu, Alexander Powers, Ron Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation, 2023.

[39] Yuejiang Yu, Chun Cai, Jiayue Wang, Zonghua Bo, Zhengdan Zhu, and Hang Zheng. Uni-dock: Gpu-accelerated docking enables ultralarge virtual screening. *Journal of Chemical Theory and Computation*, 19(11):3336–3345, 2023. PMID: 37125970.

[40] Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, Maria Paula Magarinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A Patrícia Bento, Melissa F Adasme, Peter Monecke, Gregory A Landrum, and Andrew R Leach. The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 11 2023.

[41] Wei Zhu, Miao Xu, Catherine Z Chen, Hui Guo, Min Shen, Xin Hu, Paul Shinn, Carleen Klumpp-Thomas, Samuel G Michael, and Wei Zheng. Identification of SARS-CoV-2 3CL protease inhibitors by a quantitative high-throughput screening. *ACS Pharmacol. Transl. Sci.*, 3(5):1008–1016, October 2020.