Extended Abstract

Motivation Drug discovery is a very hard and slow process, in particular due to the intractably large chemical space that has to be searched and the challenge of generating realistic drugs with desirable biological properties. A task of interest due to the potential of treating multiple diseases is to quickly develop effective, drug-like ligands to any desired protein binding pocket using computational tools. Recently, a large language model called Token-Mol Wang et al. (2024) has been proposed that efficiently generates such ligands given a pocket embedding by tokenizing the SMILES sequence and ligand geometry. However, their RL finetuning approach to generate more realistic ligands is on-policy and pocket specific, which is slow and not generalizable.

Method Token-Mol is a transformer that acts on sequences whose vocabulary contains relevant characters for SMILES strings and geometry information (torsion angles as real numbers). It is also inherently multi-modal, since it is conditioned on a protein pocket embedding generated by a pre-trained (and frozen) encoder. We build the preference dataset from CrossDocked2020 Francoeur et al. (2020) as follows: for each pocket, choose the molecule with best (more negative) binding affinity as the winning sample, and worst (less negative) binding affinity as the losing sample. We then propose to use two regularization strategies: Exact Energy Preference Optimization (E^2PO) as proposed by Gu et al. (2024) which counters winning sample overfitting, and adding an NLL loss, as proposed by Pang et al. (2024) and used in Llama 3 et al (2024). We analyze their train curves and validation curves to investigate training stability, and compare best performing models. We also perform an ablation over the DPO parameter β, which dictates how much the learned policy deviates from the reference policy.

Implementation We implement everything in Python and run all our experiments in one NVIDIA T4 GPU. The metrics we are interested in investigating are the following: binding affinity (the "strength" of binding), which is computed using QuickVina2 Alhossary et al. (2015); QED (drug-likeness), computed using RDKit; synthetic accessibility (SA) (how easy to synthesize), computed using a custom implementation from Ertl and Schuffenhauer (2009); Lipinski (oral availability), computed with RDKit; diversity (how different are the molecules for a given pocket), computed by the pairwise Tanimoto similarity between 5 ligands generated for a given pocket; and validity, the fraction of chemically valid molecules generated in a given batch. We also report the original Token-Mol "gaussian cross-entropy loss" (the "language modeling" loss) as the validation loss to track performance deviation from the original next-token prediction task.

Results Given a fixed β, we observe that even though the E^2PO regularization performs slightly better for binding affinity, NLL regularization performs better for all other drug-likeness metrics. However, training with NLL is slightly less stable and much more prone for overfitting, so it requires early stopping to achieve the best performances. From our ablations we find that $\beta = 0.1$ finds a good balance between good binding affinity and other drug-likeness metrics. Comparing to the Token-Mol baseline, the generated ligands from our best performing model has not only better average metrics, but also distributions with smaller variance, which shows it consistently generates more drug-like ligands.

Discussion One interesting observation is that, from our β ablations, it seems like binding affinity is inversely with diversity, and directly with SA. This is surprising, since intuitively we would expect lower (better) affinities to cause lower diversity, since the model is likely generating ligands more similar to the winning sample, and likely lower affinities could be caused by more artificial molecules that are higher to synthesize. This could be due to artifacts from using a very small validation set size (necessary due to limited compute), and should be investigated in studies on the full CrossDocked2020 dataset.

Conclusion In this paper, we improve on the ligand-generating language model Token-Mol by using DPO ranked by binding affinity. We find that both quantitatively and qualitatively our approach performs better than baseline, with consistently better generated ligands. Future studies should investigate if results hold or improve when trained and validated on the full CrossDocked2020 dataset.

DPOBind: Ligand Generation Through Direct Preference Optimization of Chemical Language Models

Rafael Prado Basto
Department of Computer Science
Stanford University
rbasto19@stanford.edu

Abstract

Generating effective, drug-like candidate ligands for protein binding pockets is still an open problem in structure-based drug design. Recently, machine learning methods, such as denoising diffusion and language models, have successfully been applied to the task. A shortcoming of some of these models is the challenge to not only generate ligands that will bind well, but that also have desirable drug-like properties. In this paper, we propose to improve on Token-Mol, a recent ligand-generating language model, by using Direct Preference Optimization on curated preference pairs based on the CrossDocked2020 dataset. We investigate different regularization strategies to the original DPO objective to mitigate winning sample overfitting, and perform ablation on the DPO β parameter to find the optimal setup. We find that our approach does better than baseline Token-Mol on both affinity and drug-likeness metrics, but slightly underperforms in validity of generated ligands, on a limited validation set. Future chemical language models similarly fine-tuned with DPO but with larger train and validation sizes would likely show even better performance, and are a promising direction for better structure-based drug design.

1 Introduction

Developing new drugs that are effective and will respond well in the human body is a challenging process, that takes a lot of wet-lab time and is very costly. Recently, computer-aided drug design has dramatically accelerated this process; in particular, structure-based drug design aims to generate candidate ligands that will bind to a desired protein pocket that is relevant to treating a given disease. Machine learning methods have been particularly promising for this task, due to their ability to encode protein pocket information and generate ligands with desired properties. Promising previous work has attempted to autorregressively generate ligand graphs Peng et al. (2022), or even use denoising diffusion models Guan et al. (2023). Another recent model of interest is Token-Mol Wang et al. (2024), which leverages the transformer architecture to generate a string representation (SMILES) of a candidate ligand along with geometry information in the form of torsion angles, conditioned on the protein pocket embedding. This approach is interesting since it finds a balance between the large expressiveness of modern language models with incorporating geometric information that graph-based methods excel at.

However, a shortcoming of most of these models is that the candidate ligands – even though could technically bind to the pocket – might not have chemical properties that are important to make sure the molecule can be easily synthesized, have low toxicity, or be orally available. AliDiff Gu et al. (2024) addressed this in the context of diffusion models by using Direct Preference Optimization with ligands ranked by binding affinity scores, whereas Token-Mol used a policy gradient to generate ligands with a high custom scoring function that incorporates multiple drug-likeness metrics. However, Token-Mol

finetunes only for individual protein pockets, instead of for the entire protein pocket landscape in the train set. Moreover, computing the reward function for their on-policy approach is very expensive in this case, since it involves computing binding affinities. Due to the recent good performance of DPO in language modeling tasks, in this paper, we propose to address these shortcomings of Token-Mol by using Direct Preference Optimization (DPO) Rafailov et al. (2024) to – regardless of the protein pocket – generate molecules that have better drug-likeness properties. We achieve this by, similar to Gu et al. (2024), curating a preference dataset based on binding affinity and fine tuning Token-Mol on it.

We also investigate the effect of different regularization strategies to address winning sample overfitting, and compare six drug-likeness metrics with baseline Token-Mol. Due to limited computational resources, we train and validate on a very small subset of the CrossDocked2020 dataset Francoeur et al. (2020), which contains around 2k protein-ligand pairs. Even with the small dataset size, we still observe a slight increase in overall performance from baseline. We presume that training on the full dataset would likely show more improvements, and can be a promising direction for having better and faster general-purpose models that generate effective, safe, and synthesizable ligands.

2 Related Work

Recent developments in structure-based drug designed have heavily relied on ML tools for target-aware ligand generation. An early ligand generation model by Skalic et al. (2019) relied on generative adversarial networks and a convolutional network to generate the ligand shape conditioned on the pocket structure, and uses a trained decoder to further generate the ligand SMILES. This approach is interesting, but can't be easily cast as a policy and hence it is hard to use RLHF to finetune it to better drug-like properties. Also working on 3D space, work by Peng et al. (2022) used graph neural networks to generate ligands based directly based on the pocket geometry, and autorregressively generate ligand features, but again no preference optimization is involved. Guan et al. (2023) proposed a target-aware 3D equivariant diffusion model, which generates ligands by denoising atom types, coordinates, and bonds while keeping required symmetries. Even though this model achieved great results, it similarly lacked alignment for generating ligands with desired drug-like properties. Gu et al. (2024) improves on this work by also proposing a ligand-generating diffusion model that relies on Direct Preference Optimization (DPO) to generate more drug-like ligands, which showed significant improvements. They also propose a new regularization strategy to mitigate winning sample overfitting called Exact Energy Preference Optimization (E^2PO), which shifts the winning sample distribution towards the theoretical optimum. Also attempting to regularize the DPO loss, Pang et al. (2024) proposes to add a negative log-likelihood term, which acts oppositely to E^2PO and explicitly encourages preferring the winning sample, and was successfully used by Llama 3 et al (2024).

Inspired by these methods, our approach improves on previous work by RL finetuning the ligand-generating language model Token-Mol with DPO. Token-Mol is the first of its kind in being token-only and can hence likely benefit from DPO due to its success on language modeling tasks. The different regularization strategies – both one that was successful in ligand design and another in a renowned LLM – applied to Token-Mol, which matches both descriptions, could likely generate performance gains.

3 Method

Token-Mol is a transformer language model that relies on the GPT-2 architecture. However, Token-Mol to some extent is multimodal, since its output must also depend on the protein pocket embedding. For this, they also use a custom multi-head cross-attention (which they term "condition attention"), such that generated tokens can attend to the pocket embeddings. See the picture below with the Token-Mol architecture, extracted from their paper:

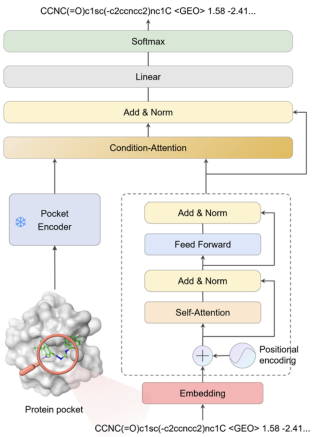


Figure 1: Token-Mol architecture, showcasing the cross-attention mechanism that allows generated tokens to attend to the pocket embedding. Source: Wang et al. (2024)

The pocket embedding is generated with the model from Zhang et al. (2023), with parameters frozen during training. For our experiments, we use the pretrained version of Token-Mol that is fine-tuned on ligand generation tasks for protein pockets. Given a pocket embedding, Token-Mol generates the ligand in the following format: "<SMILES> GEO <angle1> <angle2> <angle3> ...", where <angle> corresponds to the torsion angles that define the conformation. Their tokenizer is based on BERT with a custom vocab that contains relevant tokens for SMILES strings.

For DPO, recall that from Rafailov et al. (2024), given a preference dataset $\mathcal{D} = \{(p, m^w, m^l)\}$ with protein pockets p, and winning/losing samples m^w/m^l, respectively, they propose to optimize the following loss:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(p,m^w,m^l)\sim\mathcal{D}}[\log\sigma(\beta\log\frac{p_\theta(m^w|p)}{p_{ref}(m^w|p)} - \beta\log\frac{p_\theta(m^l|p)}{p_{ref}(m^l|p)})] \tag{1}$$

where p_θ, p_{ref} are the probabilities of generating a candidate ligand for the current and reference (base model) policies, respectively, β is a hyperparameter, and σ is a sigmoid function. In our case, $\log p(out|in)$ is computed as usual with language models: sum the log probabilities of each generated token when feeding the sequence to the model. By optimizing this loss, the model should assign higher/lower probability for winning/losing samples, respectively, and hence likely generate better ligand candidates. See the figure below for an illustration of the idea, taken from the AliDiff paper:

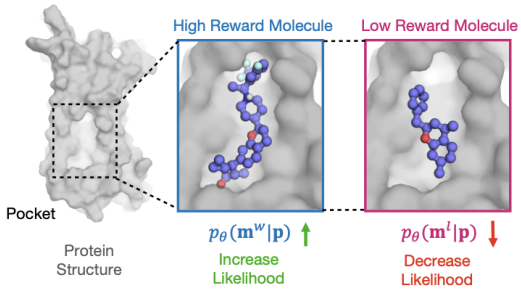


Figure 2: Illustration of the idea behind DPO in our setup. Source: Gu et al. (2024)

However, this approach is prone to overfitting to the winning sample, that is, assigning zero probability to the losing sample, which is undesirable and generates instabilities since the loss blows up. Hence, we follow the approach in Gu et al. (2024) to regularize this loss by shifting the distribution p_θ towards the optimal value of $\sigma(r^w - r^l)$, where r are the rewards. They propose the following regularized loss, termed Exact Energy Preference Optimization (E^{2}PO):

$$\mathcal{L}_{\text{E}^2\text{PO}} = -[\sigma(r^w - r^l)\mathcal{L}_{\text{DPO}} + (1 - \sigma(r^w - r^l))(1 - \mathcal{L}_{\text{DPO}})] \tag{2}$$

To study the opposite direction, inspired by Llama 3 et al (2024) and Pang et al. (2024), we also attempt to explicitly encourage the model to assign higher probabilities to the winning sample by adding a negative log-likelihood term:

$$\mathcal{L}_{\text{NLL-DPO}} = \mathcal{L}_{\text{DPO}} - \alpha \mathbb{E}_{(p,m^w,m^l)\sim\mathcal{D}} \left[\frac{\log p_\theta(m^w|p)}{|m^w|} \right] \tag{3}$$

This is interesting since Pang et al. (2024) observed significance performance gains from including this term, which helps the model learn more from the winning sample.

One thing to note is that when training, as a "validation loss" we also report the performance of the model under the original Token-Mol loss function to track if the model still does well in the base task. They propose a custom "Gaussian Cross Entropy" (GCE) loss, which instead of weighing labels from all tokens besides the ground-truth token with zero, weighs with values taken from a gaussian distribution centered around the true label. Assigning nonzero weights to neighboring tokens attempts to embed more continuity in the model and hence perform better for generating torsion angles, which are inherently continuous.

4 Experimental Setup

4.1 Dataset

We rely on a downsampled version of CrossDocked2020 Francoeur et al. (2020), which contains protein ligands cross-docked across multiple similar binding sites, which augments the usual PDBbind dataset by also including lower quality docking examples. Our filtering process is similar to Gu et al. (2024) and involves selecting docking poses that don't deviate more than 1 Angstrom from the ground truth, and proteins whose sequence identity are smaller than 30% (to ensure they are significantly different).

For the preference dataset, we have the following strategy: for each protein pocket, select the winning sample as the ligand with highest (more negative) binding affinity, and the losing sample as the one with lowest (less negative) binding affinity. Gu et al. (2024) noted that ranking samples based on binding affinity gives the best performance in their case, so we follow this approach. We end up with around 2K preference pairs for training, and 10 for validating. We realize 10 is a very small number to have any generalizable conclusions, but we had to limit the size due to the large time for docking and computing affinity of generated ligands during validation, and we can still perform interesting ablations with it.

4.2 Baselines

We compare our model with base Token-Mol, that is, Token-Mol before RL finetuning on individual pockets, since their approach is pocket specific and not generalizable for arbitrary pockets.

4.3 Evaluation Metrics

We evaluate our model on six metrics. First we compute binding affinity, that is, the free energy change when a ligand binds to a given protein pocket. The more negative the binding affinity, the stronger the binding, since it means the docked state has a lower energy. We estimate the binding affinity from the Vina score from the QuickVina2 Alhossary et al. (2015) program. Second, molecules that are drug candidates need to satisfy certain properties. To quantify a molecule's drug-likeness, we report the QED (composite score of desirable chemical properties in a drug, higher is better), Lipinski score (likelihood of being orally available, higher is better), synthetic accessibility (SA) (ease of synthesizing, higher is better), and validity (whether the SMILES string corresponds to a chemically valid molecule). Finally, for the entire validation set we compute the diversity (higher is better) of generated ligands by computing pairwise Tanimoto similarity scores between 5 generated ligands for a given pocket.

5 Results

5.1 Effect of Regularization

First, we fix $\beta = 0.1$, batch size 16, learning rate 1e-6, and investigate the effect of the different regularization strategies in both the training process and validation. See the figures below for the train/val curves. Recall that the train loss is the DPO loss, whereas the val loss is the Token-Mol GCE loss.

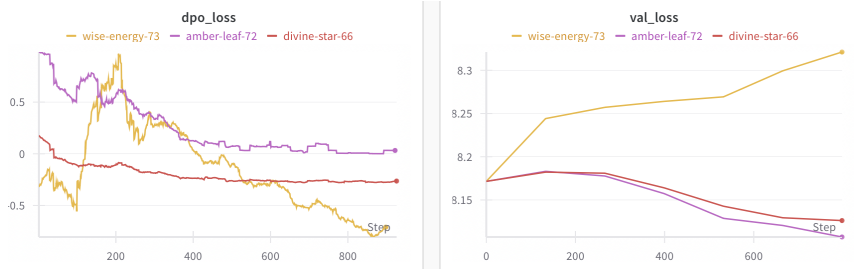


Figure 3: Train and val loss curves for non-regularized (purple), E^2PO regularization (red), and NLL regularization (yellow). We applied a moving average to smooth the plot and observe the general trends.

We observe that using the E^2PO regularization increases training stability, whereas NLL makes it much more unstable. To some extent, this makes sense, since the E^2PO loss tries to counter giving too much weight to either winning or losing samples, but the NLL loss directly encourages the model to steer towards the winning sample through a term that is unbounded (no sigmoid), and hence can likely generate less stable gradients if the probability of the winning sample is initially small, which could be the cause of the initial increase in the DPO loss of the NLL curve. As for validation, we see that both non-regularized and E^2PO regularized decrease the cross entropy loss ("GCE" loss for Token-Mol), which means the model is still doing well at next-token prediction. However, the NLL curve shows an increase, which likely means the model is indeed overfitting strongly to the winning sample, likely steering towards memorization, which could be undesirable.

We also compute the metrics of the best performing models with respect to binding affinity in each of the above training runs. See the table below:

Table 1: Performance Comparison of different regularization strategies

Regularization	Affinity (↓)	QED (↑)	SA (↑)	Lipinski (↑)	Diversity (↑)	Validity (↑)
None	-7.55	0.49	0.65	4.29	0.89	0.6
E^2PO	**-7.68**	0.49	0.65	4.27	0.89	0.6
NLL	-7.64	**0.57**	**0.67**	**4.5**	0.89	**0.8**

With regards to binding affinity, we see that E^2PO regularization does slightly better than NLL, and both do better than no regularization. For all other metrics, NLL does better than both no regularization and E^2PO, especially QED (molecule's drug-likeness), Lipinski score (oral availability), and validity, which all are good indications of molecules that are likely better drug candidates. Interestingly, the diversity is quite similar between them, which shows that NLL is actually likely not (yet) memorizing the winning sample as discussed above, which is again desired behavior. Moreover, as we see the increased validation loss of NLL and consistent decrease in DPO loss, this overfitting generates worse performance in the later iterations, so our optimal model was an early-stopped checkpoint. This shows that even though NLL regularization is more prone to overfitting, it generates better performance in the early stages of training.

5.2 β Ablations

In the DPO loss, the β hyperparameter controls how much the learned policy deviates from the reference policy, and different values of β can generate different downstream model performances. Hence, we perform an ablation over β to learn how different metrics scale with β. We use the NLL regularization with the same hyperparameters as in the previous section. See the results in the figure below:

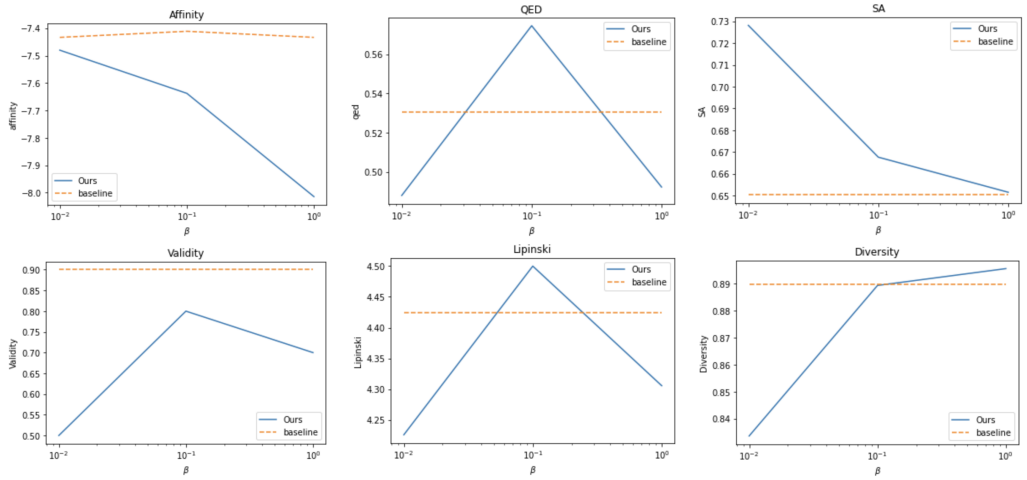


Figure 4: Beta ablations for all six studied metrics.

We see that better (more negative) affinities seem to be inversely proportional to β, whereas validity, QED, Lipinski seem to find a good balance at $\beta = 0.1$. Moreover, the SA is decreasing with β, whereas the diversity is increasing. Intuitively, we in fact would expect a different scaling, that lower β would be correlated with a lower diversity, since it would likely mean the model is predicting more of similar molecules that bind well to the site. The scaling with SA makes sense: better affinities likely mean molecules that are less realistic or harder to synthesize. Interestingly, the validity slightly increases with better affinity, which is not necessarily expected but is desired behavior from the model. One possible reason for these unexpected scaling relationships is an artifact of the very small validation set size, which is likely not large enough to highlight general trends, but they still tell something about how these metrics scale for the given proteins in the validation set.

5.3 Performance of Final Model

Using the hyperparameters of the best model from the experiments above, we get the following final metrics on the validation set, where we compare our approach with the Token-Mol baseline:

Table 2: Performance Comparison of our approach vs Token-Mol baseline

Model	Affinity (↓)	QED (↑)	SA (↑)	Lipinski (↑)	Diversity (↑)	Validity (↑)
Token-Mol Baseline	-7.45	0.53	0.65	4.42	0.89	0.9
Ours	**-7.64**	**0.57**	**0.67**	**4.5**	0.89	0.8

We see that in all metrics but validity, our model does better than the baseline, which shows that preference optimization guided by binding affinity is in fact able to improve most drug-likeness metrics. For a more thorough evaluation, we can also investigate the distribution of the metrics instead of only average values, which can give insight into how precise the model really is. See the figure below:

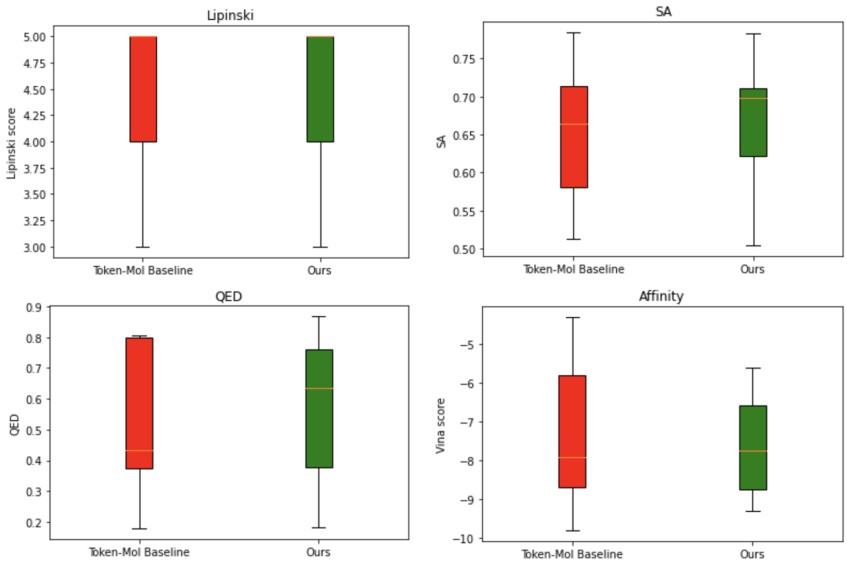
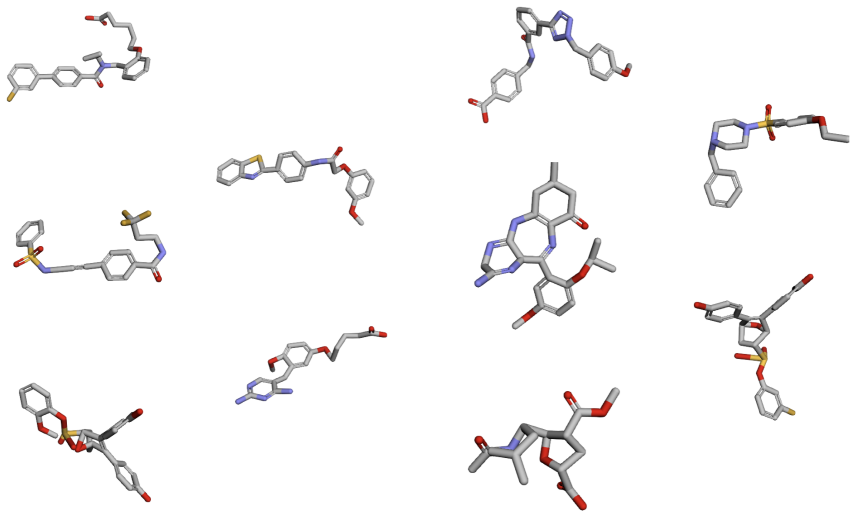


Figure 5: Distributions for four of the six metrics in the validation set (validity and diversity are numbers associated with the entire set of generated ligands). The yellow horizontal line in each box represents the median of that distribution.

We immediately see that the median values of both QED and SA have a much larger gap between our model and the baseline compared to average values. This is good, since it shows the majority of generated samples are more drug-like and more easily synthesizable. The median affinity is comparable to the baseline, but the distribution has a much smaller variance, and so does the SA distribution). This shows that the model is much more consistently generating better binding, more drug-like, and easily synthesizable candidate ligands, which is exactly what we desire from a good ligand-generating model.

5.4 Qualitative Results

For a qualitative comparison of our model with the baseline above, we sample 5 candidate ligands for the adenosine A2A receptor, which has a binding pocket that has been target for multiple therapeutic applications; in particular, recently this pocket has been investigated for treatment of Parkinson's disease, and developing good ligands for it is of significant interest. See the figure below:



(a) Ours (b) Token-Mol baseline

Figure 6: Generated ligands for ARA2A for our approach (left) and baseline (right)

From visually inspecting the molecules, we see that the ones from the baseline have some instances of multiple fused rings. Even though this can be beneficial for certain applications, it also makes synthesis harder, which is consistent with our results of larger SA for our model. Moreover, we visually see that the baseline ligands seem to have more oxygen (red) and nitrogen (blue), which can increase the number of hydrogen bond donors and acceptors, which can get it closer to violating Lipinski's rule of 5 and making it less orally available; we see this manifesting in higher Lipinski scores for our model. The set of ligands generated from our approach also contain only one molecule with a visually suboptimal conformation (atoms from different substructures too close to each other, seemingly unnatural torsion angles, etc), whereas the baseline contains two.

6 Discussion

From the experiments above, we see that fine-tuning Token-Mol on a preference dataset ranked by binding affinity using DPO showed significant overall improvements in both binding affinity and drug-likeness metrics. However, there are multiple limitations of our approach. The first obvious issue is that we trained and validated on a very small dataset, which makes it hard for the model to learn well, and our observed trends could be due to our artifacts of the small validation set. Moreover, we weren't careful in curating the validation set, so the chosen pockets will definitely not be representative of general behavior we would like a general, deployable model to have. The reason we worked with such small dataset sizes was due to limited compute resources and time, since docking and estimating binding affinity is a computationally expensive process.

However, the fact that DPO did slightly improve desired drug-likeness metrics is a good indication of the promise of this approach in chemical language models, and it could have large impacts in building faster, more effective drug design pipelines.

7 Conclusion

In this paper, we introduce using Direct Preference Optimization to improve the drug-likeness of ligands generated through the Token-Mol chemical language model for arbitrary protein binding pockets. Token-Mol previously relied in an on-policy RL finetuning approach, which is computationally expensive and pocket specific, so our innovation is making the finetuning approach general and faster through the off-policy nature of DPO. This is also beneficial since it has been shown that the contrastive nature of DPO helps with better language model alignment, and we leverage this for more drug-like ligands. We also investigate different regularization strategies to possibly mitigate winning

sample overfitting, and see that the NLL regularization used by et al (2024) performs better in early stages of training, but overfits more easily and requires early stopping. We also observe surprising trends during β ablations, such as better binding affinity scaling with ligand diversity, which we discuss is possibly an artifact of the small dataset size. However, we still observe improvements from the baseline, which is an indication that our approach was able to help generate more drug-like ligands.

For future work, it is imperative that this approach is trained and validated with the full CrossDocked2020 dataset. This will not only give the model more preference data to better learn what constitutes better or worse drug candidates, but also allow for a more thorough validation. Moreover, even though Gu et al. (2024) found better results with ranking based on binding affinity, it could be interesting to study the effect of different reward scores in the Token-Mol setting, such as using composite metrics that mix both affinity with drug-likeness metrics. Finally, it is important to validate this approach on real-world drug-discovery targets to verify the quality of the generated molecules and more thorough assessment of their therapeutic value.

8 Team Contributions

- **Rafael Prado Basto** Since this was an individual project, I was responsible for all of the project. In particular, the tasks that took a significant amount of time were downloading and processing the CrossDocked2020 dataset to generate the preference pairs, writing the DPO training script from scratch, and modifying/adapting the Token-Mol generation/logging code to support the new metrics and DPO training script.

Changes from Proposal As discussed in the milestone, my initial plan was to use Diffusion Policy and guidance to generate ligands for binding at a protein pocket. However, from better looking at the Diffusion Policy literature, I realized that implementing it in the significantly different context of molecule generation through diffusion would be impractical for a class project, since we would have to make multiple modifications to the original implementation (such as carefully reformulating the action space) and diffusion also likely would be harder to train. Hence, I pivoted to using DPO for optimizing Token-Mol on pocket-based ligand generation, as shown above.

Code availability Code is available at `https://github.com/rbasto19/cs224r_final_project.git`. Note that in my gradescope submission I only uploaded some of the main files (not the subdirectories) and training data, which are very large. The repo contains more detailed code and instructions for generating the training data.

References

Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. 2015. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics* 31, 13 (02 2015), 2214–2216. `https://doi.org/10.1093/bioinformatics/btv082` arXiv:https://academic.oup.com/bioinformatics/article-pdf/31/13/2214/49034896/bioinformatics_31_13_2214.pdf

Peter Ertl and Ansgar Schuffenhauer. 2009. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics* 1, 1 (10 Jun 2009), 8. `https://doi.org/10.1186/1758-2946-1-8`

Aaron Grattafiori et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] `https://arxiv.org/abs/2407.21783`

Paul G. Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B. Iovanisci, Ian Snyder, and David R. Koes. 2020. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *Journal of Chemical Information and Modeling* 60, 9 (2020), 4200–4215. `https://doi.org/10.1021/acs.jcim.0c00411` arXiv:https://doi.org/10.1021/acs.jcim.0c00411 PMID: 32865404.

Siyi Gu, Minkai Xu, Alexander Powers, Weili Nie, Tomas Geffner, Karsten Kreis, Jure Leskovec, Arash Vahdat, and Stefano Ermon. 2024. Aligning Target-Aware Molecule Diffusion Models with

Exact Energy Optimization. arXiv:2407.01648 [q-bio.BM] https://arxiv.org/abs/2407.01648

Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 2023. 3D Equivariant Diffusion for Target-Aware Molecule Generation and Affinity Prediction. arXiv:2303.03543 [q-bio.BM] https://arxiv.org/abs/2303.03543

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron Dror. 2021. Learning from Protein Structure with Geometric Vector Perceptrons. arXiv:2009.01411 [q-bio.BM] https://arxiv.org/abs/2009.01411

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative Reasoning Preference Optimization. arXiv:2404.19733 [cs.CL] https://arxiv.org/abs/2404.19733

Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. 2022. Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets. arXiv:2205.07249 [cs.LG] https://arxiv.org/abs/2205.07249

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290 [cs.LG] https://arxiv.org/abs/2305.18290

Miha Skalic, Davide Sabbadin, Boris Sattarov, Simone Sciabola, and Gianni De Fabritiis. 2019. From Target to Drug: Generative Modeling for the Multimodal Structure-Based Ligand Design. *Molecular Pharmaceutics* 16, 10 (07 Oct 2019), 4282–4291. https://doi.org/10.1021/acs.molpharmaceut.9b00634

Jike Wang, Rui Qin, Mingyang Wang, Meijing Fang, Yangyang Zhang, Yuchen Zhu, Qun Su, Qiaolin Gou, Chao Shen, Odin Zhang, Zhenxing Wu, Dejun Jiang, Xujun Zhang, Huifeng Zhao, Xiaozhe Wan, Zhourui Wu, Liwei Liu, Yu Kang, Chang-Yu Hsieh, and Tingjun Hou. 2024. TokenMol 1.0: Tokenized drug design with large language model. arXiv:2407.07930 [q-bio.BM] https://arxiv.org/abs/2407.07930

Odin Zhang, Jintu Zhang, Jieyu Jin, Xujun Zhang, RenLing Hu, Chao Shen, Hanqun Cao, Hongyan Du, Yu Kang, Yafeng Deng, Furui Liu, Guangyong Chen, Chang-Yu Hsieh, and Tingjun Hou. 2023. ResGen is a pocket-aware 3D molecular generation model based on parallel multiscale modelling. *Nature Machine Intelligence* 5, 9 (01 Sep 2023), 1020–1030. https://doi.org/10.1038/s42256-023-00712-7