TacoGFN: Target Conditioned GFlowNet for Structure-Based Drug Design

Tony Shen
Simon Fraser University
tsa87@sfu.ca

Mohit Pandey
University of British Columbia
mkpandey@student.ubc.ca

Martin Ester
Simon Fraser University
ester@sfu.ca

Abstract

We seek to automate the generation of drug-like compounds conditioned to target protein pocket. Most current pocket-conditioned molecule generation methods approximate the protein-molecule distribution of a finite dataset and, therefore struggle to generate molecules with significant binding improvement over the training dataset. We instead frame the pocket-conditioned molecule generation task as an RL problem and develop TacoGFN, a target conditioned Generative Flow Network model. Our method is explicitly encouraged to generate molecules with desired properties as opposed to fitting on a pre-existing data distribution. To this end, we develop transformer-based docking score prediction to speed up docking score computation and propose TacoGFN to explore molecule space efficiently. Furthermore, we incorporate several rounds of active learning where generated samples are queried using a docking oracle to improve the docking score prediction. This approach allows us to accurately explore as much of the molecule landscape as we can afford computationally. Empirically, molecules generated using TacoGFN and its variants significantly outperform all baseline methods across every property (Docking score, QED, SA, Lipinski), while being orders of magnitude faster.

1 Introduction

Structure-based drug design (SBDD) leverages target protein structures to design and optimize potential drug molecules. Due to the growing availability of protein structures from ML protein structure prediction methods [Jumper et al., 2021], and many novel targets identified from high-throughput perturbation experiments, SBDD is becoming an increasingly powerful approach in drug discovery.

Traditional SBDD uses molecular docking to screen virtual libraries of molecules for interaction with a target protein, but its efficacy is impeded by the nature of its exhaustive search within a limited virtual library. Recent works proposed accelerating virtual screening by using an ML model as the molecular docking proxy [Gentile et al., 2022] and incorporating active learning to improve molecular docking proxy model [Graff et al., 2021]. Nevertheless, one virtual screening campaign concerns a single target of interest - learning from one model does not generalize to another target.

Generative models for molecules have been proposed to more efficiently explore the chemical space, as they turn the brute-force virtual screening problem into a search problem. Current generative models condition molecule generation on 3D geometric information of the protein pocket using Geometric Deep Learning model architectures [Atz et al., 2021]. Recent works in this area typically use auto-regressive or diffusion models [Guan et al., 2023, Peng et al., 2022, Luo et al., 2022, Schneuing et al., 2023], and could theoretically generate molecule binder from a far greater chemical space than any virtual library for any given pocket.

NeurIPS 2023 Generative AI and Biology Workshop.

As most existing generative models approximate the protein-ligand distribution of the dataset, they are unable to propose molecules with significantly better binding affinities than the found in training dataset without further lead optimization. Furthermore, since the availability of such data is limited, this leads to poor generalization if the molecule complementary to a target protein pocket is outside of the training distribution.

In contrary, we learn a RL policy for exploring the chemical space of molecules and explicitly reward policies that construct molecules with high docking score, synthetic accessibility and drug-likeliness score. The performance of our method no longer depend on the size of a fixed dataset, but rather on how much compute power is available to generate molecules and compute their reward. As a result, the molecules generated have better properties than reference; the chemical space explored is no longer constrained by a fixed dataset.

In this paper, we employ the recently proposed RL method GFlowNet [Bengio et al., 2021], which constructs objects with probability proportional to their reward, thus guaranteeing a diverse set of results. We incorporate the target pocket information into GFlowNet and train a docking score prediction model based on graph transformer [Yun et al., 2020] to estimate the affinity of a molecule with respect to a given pocket. Since the initial training dataset may be small and biased, we incorporate an active learning approach to improve the generalization of the docking score prediction model. We benchmark the performance of our method, which we call TACOGFN: *Target Conditioned GFlowNet for Drug Design*, on unseen pockets against current state-of-the-art baseline methods to show that the generated molecules show higher docking scores, synthetic accessibility and drug-likeness score. In summary, the key contributions of this work are the following:

- To our best knowledge, TacoGFN is the first method that frames the pocket-conditioned molecule generation task as an RL problem - learning a policy that generates candidates with probabilities proportional to reward based on docking score and desired properties.
- To solve this problem, we propose an extension of GFlowNets to incorporate protein pocket structure context for target conditioned molecule generation.
- We incorporate an active learning approach to gradually improve the generalization of the docking score proxy, and consequently of the GFlowNet generator.
- We performed an experimental evaluation on the Cross-Docked dataset and demonstrated that TACOGFN generates molecules with better docking scores and properties compared to all existing state-of-the-art methods.

2 Related work

Structure based drug design aims to sample drug-like molecules for target protein pockets. LiGAN [Ragoza et al., 2022] uses 3D CNN to encode protein pocket structure and predict atom densities from the encoded latent space. 3DSBDD [Luo et al., 2022] and Pocket2Mol [Peng et al., 2022] builds molecule atom by atom autoregressively. Other methods such as FLAG [Zhang et al., 2023b] and DrugGPS [Zhang and Liu, 2023] build molecules fragment by fragment to leverage the chemical prior. There's also a new line of research using diffusion models [Guan et al., 2023, Schneuing et al., 2023] for SBDD. These methods typically implicitly assume an improvement in molecule affinity to pocket through generating molecules and representing pocket in 3D space. However, [Harris et al., 2023] found that despite the geometric representation, generated 3D molecules have much more physical violations and fewer key interactions compared to reference set. In our work, we generate molecules in 2D space instead to allow us to explore 1000 times more molecules compared to existing models. [1]

Goal-directed molecule generation aims to discover molecules that satisfy optimization goals such as binding activity and high drug likeliness. Reinforcement Learning (RL) methods such as ReLeaSE [Olivecrona et al., 2017], MolDQN [Zhou et al., 2019] and REINVENT [Blaschke et al., 2020] has been proposed to guide the generation of molecules toward desirable properties. MORLD [Jeon and Kim, 2020] and MoleGuLAR [Goel et al., 2021] combine RL and docking calculations to design novel binders. Recently, MOOD [Lee et al., 2023] incorporates out-of-distribution and

[1] Our method generates and evaluates $\sim$ 100M molecules pocket complexes compared to $\sim$ 100k complexes from CrossDock dataset used by existing methods.

property-guided exploration in diffusion models for goal-directed molecular generation. Existing goal-directed methods are constrained by a fixed objective function. They can only be trained to generate high-affinity molecules for a single target, and therefore, a new model is needed for each different protein pocket. In contrast, TACOGFN is conditioned on the protein pocket structure itself and is trained to generate high-affinity molecules for any given pocket structure.

Generative Flow Network (GFlowNet, GFN) [Bengio et al., 2021] learns a stochastic policy for generating an object (like a molecular graph) from a sequence of actions. GFlowNet learns a policy such that the probability of generating an object is proportional to a given reward for that object, therefore generates a more diverse set of solutions compared to other RL methods. Following [Jain et al., 2023a], we incorporate an active learning algorithm and offline policy training with GFlowNet, taking advantage of molecular docking as our oracle. We also use docking oracle queried batches as offline policy training for GFlowNet, to guarantee explorations around known high reward regions. We use Multi-objective GFlowNets [Jain et al., 2023b] to generate molecules optimizing for both docking score, drug-likeliness, and synthesizability.

3 Method

Our goal is to train a single conditional generative model that generalizes over protein pockets. We first pre-train the pocket embedding using geometric-vector-perceptron-based graph neural network (GVP-GNN) [Jing et al., 2021] on the docking score prediction task. We then take these learned graph-level embedding of the protein pocket as the conditioning for TACOGFN.

3.1 Target pocket conditioned GFlowNet

We introduce target pocket context as the condition for GFlowNet to generate molecules with preferential interaction with specific protein pockets, building on multi-objective GFlowNet conditioning in [Bengio et al., 2023]. The goal is to learn a single target-conditioned GFlowNet that models distribution associated with all pocket structures respectively. While many pocket-ligand complex representation methods that take advantage of geometric deep learning exist, the memory and time complexity associated with using these models for predicting every GFlowNet action makes exploring a large number of molecules computationally expensive. Instead, we use a pocket encoder trained from the docking score prediction task (with weights frozen) to pre-compute the pocket embedding to condition molecule generation. This allows GFlowNet to leverage a pocket representation suitable for computing pocket-ligand interaction, to condition the generation of high affinity molecules.

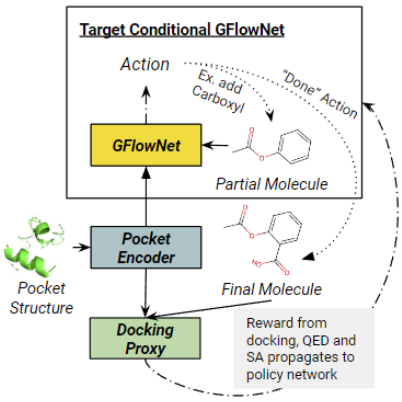

Figure 1: Our proposed approach consists of two main components: (1) GFlowNet generator (yellow box) which is conditioned with target pocket embedding from the pocket encoder (cyan box) (2) Docking score predictor (green box) that takes the pocket embedding and generated molecule to compute a docking reward to train the GFlowNet policy.

3.2 Docking score proxy

Docking 100 million compound to protein pocket at the standard rate of 10 s per compound on a single CPU core will take more than 3 years. To incorporate the docking score as a reward that is fast to compute, we propose a docking score predictor as the proxy for the molecular docking program. The model is trained in an end-to-end manner. First, the pocket graph embedding is computed using a GVP-GNN [Jing et al., 2021] as the pocket encoder. The graph transformer then takes in the molecular graph and the pocket graph embedding as inputs and extracts a ligand-pocket complex embedding. This embedding is passed through MLP to obtain a docking score prediction.

3.3 Active learning and offline policy

As GFlowNet often samples molecules beyond the initial training distribution of the docking score predictor, the predictor can be overconfident in assigning high rewards for out-of-distribution molecules. We incorporate a similar active learning scheme from [Jain et al., 2023a], leveraging the docking program to compute new training labels used for improving the generalization of the docking proxy. In addition, the evaluated batch from active learning is added to GFlowNet's offline training dataset, to ensure exploration around known high reward regions.

4 Experiments

Dataset and evaluation metrics Following the same data preparation and pocket sequence-based splitting as [Peng et al., 2022] and [Guan et al., 2023], we obtain 100k pocket-ligand complexes from Cross-Docked dataset [Francoeur et al., 2020]. We identify 15207 unique pockets and 91916 unique ligands within these complexes. Subsequently, we compute the docking score using QVina2 [Alhossary et al., 2015] for all these complexes. The pocket-ligand pairs and associated docking score form the initial dataset for training the docking score predictor.

We evaluate the performance of target conditional molecule generation using widely used metrics from previous work [Peng et al., 2022, Luo and Ji, 2022, Guan et al., 2023, Zhang and Liu, 2023]. (1) **Vina Score** approximates binding affinity between molecules and their target pockets; (2) **High Affinity** measures the percentage of generated molecules with higher affinity than the reference molecule; (3) **QED** measures how closely a molecule resembles properties of bioactive compounds present in PubChem; (4) **SA** (synthetic accessibility) estimates how easily the molecule can be synthesized; The score is normalized to a range of 0 to 1 using the formula $(10 - SA)/9$. (5) **Lipinski** measures the number of rules satisfied in Lipinski rule five [Lipinski et al., 1997] - a heuristic measuring drug-likeness; (6) **Diversity** is the average pairwise fingerprint dissimilarity between generated molecules for each target; (7) **Inference Time** is the average time in seconds to generate 100 molecules for one target pocket.

Baselines and evaluation We compare our method without active learning (TACOGFN) with LiGAN [Ragoza et al., 2022], TargetDiff [Guan et al., 2023], DiffSBDD [Schneuing et al., 2023], Pocket2Mol[Peng et al., 2022] and DrugGPS[Zhang and Liu, 2023] in the fixed dataset setting. Since there is currently no established baseline for target-conditioned generation with active learning, TACOGFN-AL is only compared against the original CrossDocked dataset. The TOP50 variant uses the same model as TACOGFN-AL, but with an added fast filtering step. We rank 100 molecules based on our trained *docking score proxy* and keep only the top 50 molecules with the highest docking scores, excluding those with QED scores below 0.4 for each pocket.

Following previous works, we sample 100 molecules from the target conditional GFlowNet for each pocket. For TACOGFN (fixed dataset setting), the docking score predictor is trained on the Cross-Docked dataset and used for providing the reward for target conditional GFlowNet generation. In the active learning setting (TACOGFN-AL), we conduct 3 rounds of such training and query the docking oracle for 30k training samples each round.

Results TACOGFN outperforms all baselines in generating molecules with higher scores for all molecular properties, including Vina Score. In particular, TACOGFN achieves excellent QED (0.681) compared to the previous best (0.592). The poor QED of previous methods is likely due to the low QED of the training dataset and their reliance on maximizing the likelihood. Under our RL framework, molecules generated from TACOGFN have significantly better properties compared to the training dataset. Our model exhibits lower diversity compared to other methods, possibly because we construct molecules fragment by fragment from a limited vocabulary, rather than atom by atom. In the active learning setting, TACOGFN-AL further improves the Vina score to -7.678 compared to -7.405 in the base model. A quick post-filtering of generated molecules using the trained docking proxy in TACOGFN-AL-TOP50 improves the Vina score to an impressive -7.924. Finally, due to the simple yet effective pocket representation and molecule construction process, TACOGFN generates molecules 50 to 1000 times faster compared to baseline methods.

Table 1: Evaluation of generated molecules for targets from the Cross-Docked test set. The baseline results are taken from the corresponding publications. * denotes values not provided by the authors. We report the means and standard deviations, aggregated from the metrics for each test pocket.

	Methods	Vina Score	High Affinity	QED	SA	Lipinski	Diversity	Time
Fixed Dataset	Reference	-7.158±2.10	-	0.484±0.21	0.732±0.14	4.367±1.14	-	-
	LiGAN	-6.114±1.57	0.238±0.28	0.369±0.22	0.590±0.15	4.027±1.38	0.654±0.12	*
	Pocket2Mol	-7.288±2.53	0.542±0.32	0.563±0.16	0.765±0.13	4.902±0.42	0.688±0.14	2503.5±2207
	TargetDiff	-7.318±2.47	0.581±*	0.483±0.20	0.584±0.13	4.594±0.83	0.718±0.09	$\sim$ 3428
	DiffSBDD	-7.333±2.56	*	0.467±0.18	0.554±0.12	4.702±0.64	**0.758±0.05**	160.3 ± 73.3
	DrugGPS	-7.345±2.42	0.620±0.29	0.592±0.21	0.728±0.23	4.923±0.11	0.695±0.17	956.3±451.6
	TacoGFN	**-7.405±1.70**	**0.625±0.35**	**0.681±0.20**	**0.783±0.07**	**4.938±0.24**	0.653±0.07	**2.90±0.28**
AL	TacoGFN-AL	-7.678±1.71	0.706±0.31	0.640±0.21	0.814± 0.06	4.931±0.26	0.663±0.07	3.07±0.31
	TacoGFN-AL-*Top50*	**-7.924±1.55**	**0.771±0.31**	**0.678±0.15**	**0.821±0.06**	**4.997±0.06**	**0.675±0.07**	3.07±0.31

5 Conclusions

We presented TACOGFN - a GFlowNet tailored to the task of generalizable structure-based drug design. This represents a paradigm shift from previous generative approaches, which learn the distribution of a fixed dataset, to an RL approach where the model is rewarded for exploring the chemical space and generating molecules with desired interactions and properties. To search for high-affinity molecules with desired properties in the vast chemical space efficiently, we design a target-conditioned GFlowNet, which incorporates pocket context and gets rewards from our proposed docking score proxy. Empirically, we show that TACOGFN outperforms the state-of-the-art on target conditional molecule generation for all molecular property metrics, while taking orders of magnitude less time. We show the incorporation of active learning and screening using docking score proxy on generated molecules can further facilitate the discovery of molecules with desirable properties. In summary, TACOGFN and its variants can offer great value for many existing real-world SBDD campaigns. Interesting future directions include leveraging uncertainty estimation in docking proxy, exploring better ways of pocket representation, generating molecules using known reactions [Gao et al., 2022] and adopting continuous GFlowNet [Lahlou et al., 2023] for molecule structure and conformation co-design.

Acknowledgments and Disclosure of Funding

We thank Emmanuel Bengio and Shuman Peng for the valuable discussions and feedback. This work was supported by the NSERC Discovery grant.

References

Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 02 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv082. URL `https://doi.org/10.1093/bioinformatics/btv082`.

Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, 3(12):1023–1032, dec 2021. doi: 10.1038/s42256-021-00418-8. URL `https://doi.org/10.1038/s42256-021-00418-8`.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations, 2023.

Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. REINVENT 2.0: An AI tool for de novo drug design. *Journal of Chemical Information and Modeling*, 60(12):5918–5922, October 2020. doi: 10.1021/acs.jcim.0c00915. URL `https://doi.org/10.1021/acs.jcim.0c00915`.

Paul G. Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B. Iovanisci, Ian Snyder, and David R. Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of Chemical Information and Modeling*, 60(9):4200–4215, aug 2020. doi: 10.1021/acs.jcim.0c00411. URL https://doi.org/10.1021/acs.jcim.0c00411.

Wenhao Gao, Rocío Mercado, and Connor W. Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design, 2022.

Francesco Gentile, Jean Charle Yaacoub, James Gleave, Michael Fernandez, Anh-Tien Ton, Fuqiang Ban, Abraham Stern, and Artem Cherkasov. Artificial intelligence–enabled virtual screening of ultra-large chemical libraries with deep docking. *Nature Protocols*, 17(3):672–697, February 2022. doi: 10.1038/s41596-021-00659-2. URL https://doi.org/10.1038/s41596-021-00659-2.

Manan Goel, Shampa Raghunathan, Siddhartha Laghuvarapu, and U. Deva Priyakumar. MoleGuLAR: Molecule generation using reinforcement learning with alternating rewards. *Journal of Chemical Information and Modeling*, 61(12):5815–5826, December 2021. doi: 10.1021/acs.jcim.1c01341. URL https://doi.org/10.1021/acs.jcim.1c01341.

David E. Graff, Eugene I. Shakhnovich, and Connor W. Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chem. Sci.*, 12:7866–7881, 2021. doi: 10.1039/D0SC06805E. URL http://dx.doi.org/10.1039/D0SC06805E.

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 *International Conference on Learning Representations*, 2023.

Charles Harris, Kieran Didi, Arian R. Jamasb, Chaitanya K. Joshi, Simon V. Mathis, Pietro Lio, and Tom Blundell. Benchmarking generated poses: How rational is structure-based drug design with generative models?, 2023.

Moksh Jain, Emmanuel Bengio, Alex-Hernandez Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ekbote, Jie Fu, Tianyu Zhang, Micheal Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with gflownets, 2023a.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets, 2023b.

Woosung Jeon and Dongsup Kim. Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors. *Scientific Reports*, 10(1), December 2020. doi: 10.1038/s41598-020-78537-2. URL https://doi.org/10.1038/s41598-020-78537-2.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-García, Léna Néhale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks, 2023.

Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.

Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pages 18872–18892. PMLR, 2023.

Christopher A. Lipinski, Franco Lombardo, Beryl W. Dominy, and Paul J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 23(1):3–25, 1997. ISSN 0169-409X. doi: https://doi.org/10.1016/S0169-409X(96)00423-1. URL https://www.sciencedirect.com/science/article/pii/S0169409X96004231. In Vitro Models for Selection of Development Candidates.

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

Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=C03Ajc-NS5W.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets, 2022.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1), September 2017. doi: 10.1186/s13321-017-0235-x. URL https://doi.org/10.1186/s13321-017-0235-x.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

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

Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chem Sci*, 13:2701–2713, Feb 2022. doi: 10.1039/D1SC05976A.

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.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. Graph transformer networks, 2020.

Yangtian Zhang, Huiyu Cai, Chence Shi, Bozitao Zhong, and Jian Tang. E3bind: An end-to-end equivariant network for protein-ligand docking, 2023a.

Zaixi Zhang and Qi Liu. Learning subpocket prototypes for generalizable structure-based drug design, 2023.

Zaixi Zhang, Yaosen Min, Shuxin Zheng, and Qi Liu. Molecule generation for target protein binding with structural motifs. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=Rq13idF0F73.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N. Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific Reports*, 9(1), jul 2019. doi: 10.1038/s41598-019-47148-x. URL https://doi.org/10.1038%2Fs41598-019-47148-x.

Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, Chang Ma, Runcheng Liu, Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Torchdrug: A powerful and flexible machine learning platform for drug discovery, 2022.

Appendix

In this study, we used the open-sourced code for GFlowNet Bengio et al. [2021] and E3Bind Zhang et al. [2023a]. Our models were implemented using the Pytorch Paszke et al. [2019] and TorchDrug Zhu et al. [2022] libraries, which enabled efficient training and evaluation. We utilized RDKit Landrum et al. [2006], a widely-used chem-informatics library, to handle the molecular structures and compute chemical properties.

Running all experiments presented in this paper takes an estimated 2.5 GPU days on a single Nvidia RTX 3090. This includes the time for training and inference of the docking score proxy. Computing the docking score for evaluation and active learning takes about 3 days on a 6-core CPU.

Additional GFlowNet details

GFlowNet background We follow Bengio et al. [2021] to sequentially construct a molecule fragment by fragment. Graph transformer is used to implement GFlowNet for sampling the next action. State s_0 is initialized as an empty molecular graph. For each step, we sample an action a conditioned on state s and context c and obtain the next state by applying the action on the previous state through transition function $s' = T(a, s)$. The state and action space can be represented by a flow network $N = (S, E)$. Where each (partial) state $s \in S$ corresponds to a node in the flow network, each transition $s \to s' \in E$ corresponds to an edge connecting the nodes. The flow network is required to be directed and acyclic, which means actions are constructive and cannot be undone. A special action indicates the molecule object is complete. The construction of an object x is defined as a trajectory of states $\tau = (s_0 \to s_1 \to \cdots \to x)$.

Loss function We adopt the trajectory balance loss Malkin et al. [2022], a loss function for GFlowNet that speeds up learning speed through more efficient credit assignment and robustness to long trajectories and large vocabularies. We assess the loss of an action trajectory τ within the context of a pocket c. The GFlowNet model, characterized by parameters θ, generates forward policy estimates denoted as P_{F_θ}. The reward function for the end state molecule x given the pocket c is represented by R. Additionally, Z_θ is a neural network tasked with estimating $F(s_0)$. Overall, the loss function is defined as:

$$L_{TB}(\tau, c; \theta) = \left(\log \frac{Z_\theta(c) \prod_{s \to s' \in \tau} P_{F_\theta}(s'|s, c)}{R(x|c)} \right)^2 \tag{1}$$

Target conditioning For each training trajectory, a random protein pocket is drawn from the CrossDocked training set as the target context. The pocket embedding is fed to the graph transformer responsible for sampling the next action as an additional virtual node, in addition to the multi-objective conditioning virtual node Jain et al. [2023b]. Upon completion, the docking score is predicted for the molecule with respect to the chosen pocket context.

Active learning and offline policy details

Active Learning For each round, we use the trained GFlowNet policy to sample 2 molecules for each of the pockets from the CrossDocked training set. We sample a total of 30k molecules per round for the 15k pockets. After computing the ground truth docking score for these molecule pocket pairs, they are added to the dataset used by the docking score proxy and GFlowNet's offline policy training.

Offline Policy Given a molecule m from training set, we sample a trajectory $\tau = (m, s_n, \cdots, s_0)$ that leads to m from s_0. To find a parent state of s', we sample from a uniform probability of the state space $X \subset S$, where $s \to s'$ if $s \in X$. Let $\gamma \in [0, 1)$ be a hyper-parameter indicating the proportion of offline training. At each step of GFlowNet training, $\gamma * BatchSize$ number of trajectories will be sampled from the training dataset, and the rest of the trajectories generated from the current online policy.

CrossDock as offline data The initial CrossDocked dataset is used as offline data of both TACOGFN and TACOGFN-AL. However, we were only able to map about 10% of the molecules

from the CrossDocked dataset into a graph compatible with GFlowNet's fragment-based molecular construction environment. It's conceivable that the performance of TACOGFN under the no active learning setting will improve further if all molecules from the existing dataset can be used as offline data. We plan to address the development of an enhanced algorithm or molecular construction environment for calculating trajectories of existing molecules in future research.

Reward details

Processing rewards We compute the reward as a weighted sum of the predicted docking score, drug-likeliness score (QED) and synthetic accessibility (SA), all transformed to a score from 0 to 1. Docking score is typically a negative value that represents binding energy. Since a lower docking score value is preferred, we clip the value to the interval [-15, 0], take its opposite, and normalize it to the interval [0, 1]. The best docking score observed in the CrossDocked dataset is around -15 kcal/mol, so -15 is chosen as the lower bound. A positive docking score indicates non-existent binding, so we set all positive scores to have a 0 reward for docking. QED is already a value from 0 to 1. SA is normalized by the formula $(10 - SA)/9$.

Scalarization The scalarization method from Jain et al. [2023b] is used to transform d objectives into a single reward $R = \sum_{i=1}^{d} w_i R_i$, via weights vector $\mathbf{w}$, where $w_i > 0$ and $\sum_{i=1}^{d} w_i = 1$. During training, these reward weights are drawn from the Dirichlet distribution of order d with parameter $\alpha = (1, 1, \cdots, 1)$.

Molecular domain details

Available fragments We use the same fragment based molecular construction environment as Bengio et al. [2021]. There is a library of 72 pre-defined fragments. The 72 fragment in SMILES format are `Br`, `C`, `C#N`, `C1=CCCCC1`, `C1=CNC=CC1`, `C1CC1`, `C1CCCC1`, `C1CCCCC1`, `C1CCNC1`, `C1CCNCC1`, `C1CCOC1`, `C1CCOCC1`, `C1CNCCN1`, `C1COCCN1`, `C1COCC[NH2+]1`, `C=C`, `C=CC`, `C=N`, `C=O`, `CC`, `CC(C)C`, `CC(C)O`, `CC(N)=O`, `CC=O`, `CCC`, `CCO`, `CN`, `CNC`, `CNC(C)=O`, `CNC=O`, `CO`, `CS`, `C[NH3+]`, `C[SH2+]`, `Cl`, `F`, `FC(F)F`, `I`, `N`, `N=CN`, `NC=O`, `N[SH](=O)=O`, `O`, `O=CNO`, `O=CO`, `O=C[O-]`, `O=PO`, `O=P[O-]`, `O=S=O`, `O=[NH+][O-]`, `O=[PH](O)O`, `O=[PH]([O-])O`, `O=[SH](=O)O`, `O=[SH](=O)[O-]`, `O=c1[nH]cnc2[nH]cnc12`, `O=c1[nH]cnc2c1NCCN2`, `O=c1cc[nH]c(=O)[nH]1`, `O=c1nc2[nH]c3ccccc3nc-2c(=O)[nH]1`, `O=c1nccc[nH]1`, `S`, `c1cc[nH+]cc1`, `c1cc[nH]c1`, `c1ccc2[nH]ccc2c1`, `c1ccc2ccccc2c1`, `c1ccccc1`, `c1ccncc1`, `c1ccsc1`, `c1cn[nH]c1`, `c1cncnc1`, `c1cscn1`, `c1ncc2nc[nH]c2n1`.

Available actions There are 3 types of actions: *adding a fragment*, *setting the connection point*, and *stop*. *Adding a fragment* adds a new fragment node, and a new edge between the new node and an existing node (if the graph is not previously empty). *Setting the connection point* adds the edge feature which specifies which two atoms, one from each of the connected fragments, will have a single bond added between them. Note that the edge feature can only be set once per edge to prevent cycles in the flow graph. *Stop* marks the finish of a molecule. We set an upper limit of 9 fragments for each graph. Illegal actions that produce invalid molecules are masked out.

Additional experimental details

Processing positive docking score On the rare occasion that the GFlowNet generation produces a molecule with a non-existent binding as indicated by a positive docking score, we set its Vina Score as 0 and include this data point in our calculation. In practice, only 0.5% (50 out of 10,000) molecules generated during evaluation had a positive docking score.

Reward scaling Each reward can be scaled by a constant factor - the reward scale to either increase or decrease that reward's importance in the total reward. Theoretically, multi-objective GFlowNet can learn to generate molecules on the entire Pareto front without the need to scale any reward. However, in practice, scaling the reward score by a constant factor larger than 1 enables the model to generate molecules with higher scores for that particular property. Reward scaling is used to upscale the importance of QED and SA to ensure the model places less importance on the less certain predicted docking score. The chosen reward scales for each objective are listed in Table 2. We leave the research direction of using low-fidelity reward signals in GFlowNet for future work.

Evaluation reward weight distribution Unlike existing methods that model the distribution of molecule given target pocket, TACOGFN models molecule distribution conditioned to target pocket and multi-objective reward preference. By modifying the reward preference vector, TACOGFN is able to model a family of distributions and generate molecules on various points of the Pareto front. The choice for reward preference vector depends on the specific application's prioritization of the objectives. In our case, we picked the α of the Dirichlet process for reward weight sampling such that the molecules generated simultaneously had a higher average Vina Score, QED, and SA than all existing methods. The chosen α values are listed in Table 2. It was possible to generate a higher Vina Score at the cost of lower QED, and vice versa. We leave the evaluation of Pareto front related metrics for TACOGFN in future work.

Sample of generated molecules

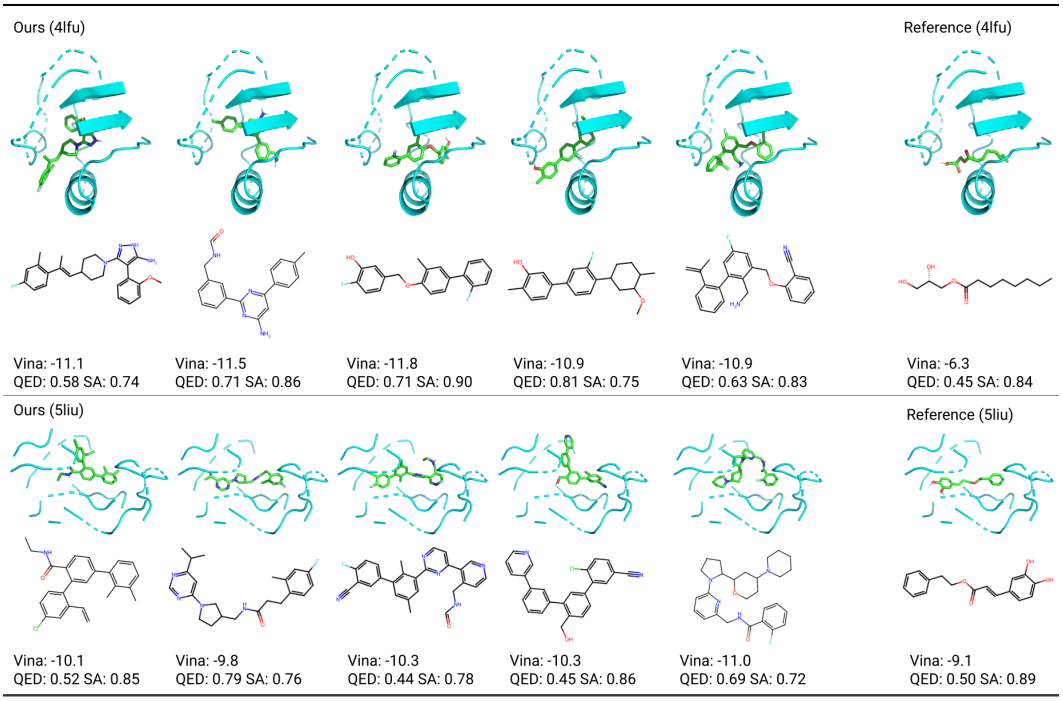

Figure 2: Examples of generated molecules from TACOGFN with higher docking scores than the references. We report the vina scores, QED, and SA scores for each molecule. The target protein pocket is visualized as Ribbon diagrams.

Algorithms

We summarize the algorithms for active learning loop and docking score prediction here.

Algorithm 1 Active Learning with GFlowNet

Input:
O: Docking program to evaluate protein pocket P_i and ligand L_i and return labels Y_i
M: Docking score predictor that predicts docking score using protein pocket P_i and ligand L_i
R: Computes reward from predicted docking, QED, SA for ligand L_i and a given protein pocket
T: Generative policy trainable from a reward function R and from which candidates m can be sampled
$D_1 = \{(P_i, L_i, y_i)\}$: Initial dataset with $y_i = O(P_i, L_i)$
N: Number of active learning rounds (outer loop iterations)
b: Size of candidate batch to be generated
Algorithm:
for $i = 1$ to N **do**
  Fit M on dataset D_i
  Train GFlowNet T using reward function R
  Sample query batch $B = \{m_1, \ldots, m_b\}$ using T
  Evaluate batch B with O: $D_{i+1} = \{(m_i, O(m_i))\}$
  Update dataset $D_{i+1} = D_i \cup D_{i+1}$.
end for

Algorithm 2 Docking Score Prediction

Input:
$p = (\mathbf{r}_i, \mathbf{c}_i)_{i=1}^n$: protein pocket with n amino acids; its i-th residual has type $\mathbf{x}_i$ with coordinate $\mathbf{c}_i$
$m = \{A, B\}$: molecule represented by 2D graph; its atom has element type and edge has bond features.
Output:
y: predicted docking score of the protein pocket and the molecule.
Algorithm:
$\{h_j^p\} = GVPGNN(p)$
$h^p = Mean(\{h_j^p\})$
$\{h_i^c\} = GraphTransformer(m, h^p)$
$y = MLP(Mean(\{h_i^c\}))$
return y

Hyperparameters

Table 2: Hyperparameters used for target conditional GFlowNet

Hyperparameters	Values	
	TacoGFN	TacoGFN-AL
Batch size	384	384
Reward exponent	$\beta \sim Uniform(0, 96)$	$\beta \sim Uniform(0, 96)$
Reward exponent for evaluation/AL	96	96
Reward scale - QVina	1	1
Reward scale - QED	2	2
Reward scale - SA	1.25	1.25
Number of training steps	7000	7000
Number of transformer layers	6	6
Number of transformer heads	2	2
Transformer node embedding size	128	128
Learning rate for GFN's P_F	10^{-4}	10^{-4}
Learning rate for GFN's $Z-$estimator	10^{-3}	10^{-3}
Sampling moving average τ	0.95	0.95
Random action probability	0.01	0.01
Offline ratio	0.25	0.25
# of molecules sampled/pocket/round	-	2
Evaluation dirchlet α_{QVina}	5	5
Evaluation dirchlet α_{QED}	0.5	0.5
Evaluation dirchlet α_{SA}	2.25	2.25

Table 3: Hyperparameters used for docking score proxy

Hyperparameters	Values
Epoch (per round)	25
Batch size	16
Learning rate	10^{-4}
Number of transformer layers	3
Number of transformer heads	2
Transformer node embedding size	128