SynFlowNet: Design of Diverse and Novel Molecules with Synthesis Constraints

Miruna Cretu[1], **Charles Harris**[1], **Ilia Igashov**[2], **Arne Schneuing**[2], **Marwin Segler**[3],
Bruno Correia[2], **Julien Roy**[4], **Emmanuel Bengio**[4], **Pietro Liò**[1]
[1]University of Cambridge, [2]EPFL, [3]Microsoft Research, [4]Valence Labs
`{mtc49, cch57, pl219}@cam.ac.uk`,
`emmanuel.bengio@recursionpharma.com`

Abstract

Generative models see increasing use in computer-aided drug design. However, while performing well at capturing distributions of molecular motifs, they often produce synthetically inaccessible molecules. To address this, we introduce SynFlowNet, a GFlowNet model whose action space uses chemical reactions and buyable reactants to sequentially build new molecules. By incorporating forward synthesis as an explicit constraint of the generative mechanism, we aim at bridging the gap between in silico molecular generation and real world synthesis capabilities. We evaluate our approach using synthetic accessibility scores and an independent retrosynthesis tool to assess the synthesizability of our compounds, and motivate the choice of GFlowNets through considerable improvement in sample diversity compared to baselines. Additionally, we identify challenges with reaction encodings that can complicate traversal of the MDP in the backward direction. To address this, we introduce various strategies for learning the GFlowNet backward policy and thus demonstrate how additional constraints can be integrated into the GFlowNet MDP framework. This approach enables our model to successfully identify synthesis pathways for previously unseen molecules.

1 Introduction

Designing molecules with targeted biochemical properties is a critical challenge in drug discovery, where computational models could play a significant role to increase efficiency and effectiveness. Recently, generative models have lead to a renaissance in *de novo* molecular design (Stanley & Segler, 2023; Du et al., 2024). However, most current *de novo* design models do not explicitly account for synthetic accessibility.

In this work, we introduce SynFlowNet, a GFlowNet specifically trained to generate molecules from documented chemical reactions and purchasable starting materials, thereby constraining exploration to a synthetically accessible chemical space and sampling not only target compounds but also the synthetic routes leading to them. Our main contributions are:

- We train a GFlowNet using an action space defined by documented chemical reactions and purchasable starting materials to generate synthesizable molecules.
- We show the advantage of using a reaction-based environment over a fragment-based one in terms of synthesisability, and the benefits of GFlowNets over Reinforcement Learning (RL) in terms of sample diversity on several targets. Additionally, when comparing to other generative methods with synthetically-accessible outputs, we show that SynFlowNet generates more novel candidates w.r.t known drug-like molecules.

Source code available at `https://github.com/mirunacrt/synflownet`

Machine Learning for Structural Biology Workshop, NeurIPS 2024.

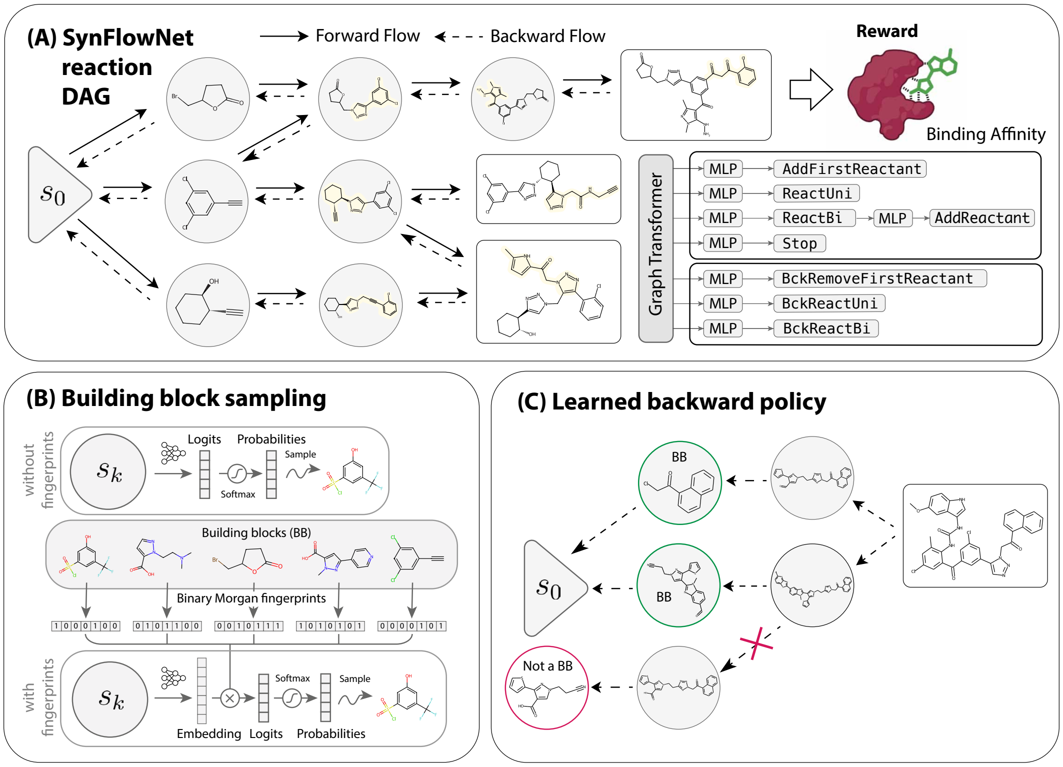

Figure 1: **SynFlowNet allows for synthesis-aware molecule generation.** (A) The state space is induced by combining purchasable building blocks (BBs) and chemical reactions. Every final molecule (rectangle box) is associated with a reward. Training trajectories are constructed by sampling the model forward. Our policy $P_F(a|s)$ is parameterized as a graph transformer which at each timestep processes the current molecular state s_t and outputs a shared embedding which is passed to separate MLP heads to predict the action logits for different action types (5 forward and 3 backward action types). An action a_t is then sampled from this hierarchical distribution to transition to the next state via a reaction. Newly added BBs are highlighted in yellow. (B) To allow handling large sets of BBs (up to 200k), we represent them using Morgan fingerprints and compute the probability of sampling a particular BB from the normalised dot product between this representation and the embedding vector of the current state. (C) Finally, when traversing the MDP backwards, to reduce the probability of exiting the MDP defined by our set of reactions and BBs, we train the backward policy to avoid paths that do not terminate in the initial state s_0.

- We evaluate different action representation alternatives to allow efficient scaling of the action space to up to 200k chemical building blocks.
- We identify an inherent problem of employing a reaction-based Markov Decision Process (MDP) with GFlowNets, stemming from the lack of guarantee that backward-constructed trajectories can return to s_0. To resolve this issue, we propose one of the first attempts at training the backward policy in a GFlowNet with a separate objective from for the forward policy, which we show to correct backward-generated trajectories and improve sample quality and diversity.
- We show that our proposed framework can be integrated with target-specific experimental data to inform selection of building blocks, which improves the efficiency of our model.

2 Methods

In this work, we present a framework to train GFlowNets on a Markov Decision Process (MDP) made of molecules obtained from sequences of chemical reactions. Below, we describe how this compositional space of synthesizable molecules is assembled (2.1), we present a method making use of backward policies to palliate imperfect information contained in reaction templates (2.2) and present the model used to navigate that state space and learn the target distribution (C.1).

2.1 MDP of chemically accessible space

Problem definition We model synthetic pathways as trajectories in a GFlowNet, starting from purchasable compounds and ending with molecules that are optimized for some desired properties, via a set of permissible reaction templates. At each timestep t, the state s_t represents the current molecule and stepping forward in the environment consists in building up the molecule by applying new pairs of reactions and reactants until either a termination action is chosen or the path reaches a maximum length. We encode reaction templates using RDKit reaction SMARTS (see Figure A.1).

Forward Action Space We define five types of forward actions: `Stop`, `AddFirstReactant`, `ReactUni`, `ReactBi`, and `AddReactant`. `ReactUni` and `ReactBi` represent uni-molecular and bi-molecular reactions. The `AddReactant` action, which is available only after a `ReactBi` action, represents the choice of reactant for the bi-molecular reaction. In more detail, each trajectory starts from an empty molecular graph which is followed by a building block sampled from `AddFirstReactant`. We then continue based on the sampled action type as follows: (a) if the action type is `Stop`, we reach a terminal state and end the trajectory; (b) if a `ReactUni` action is sampled, we apply the uni-molecular reaction template to the molecule in state s to yield the product in state s'; (c) if the action type is `ReactBi`, the sampled reaction is used as input to an additional MLP, together with the state embedding, to sample a subsequent action of type `AddReactant`.

Backward Action Space The GFlowNet framework requires us to model travelling *backward* along trajectories in the state-space. To unfold a reverse trajectory we proceed similarly: (a) if the action-type is a `BckReactUni`, the action yields the reactant molecule directly; (b) if the action type is `BckReactBi`, we obtain two reactants, and the molecule that is **not** a building block becomes the *next* state (or *previous* state in the DAG). If the two resulting reactants are both building blocks (which happens at the beginning of the forward trajectory), the molecule that populates the next state is picked with $p = {}^{1}\!/_{2}$ from the two building blocks. The last action in a backward trajectory is `BckRemoveFirstReactant`, leading to the empty molecular graph s_0 (initial state).

2.2 Challenges with Generating Backward Paths

Given the synthesis pointed DAG $\mathcal{G} = (\mathcal{S}, \mathcal{A})$, one needs to define a forward probability function P_F and a backward probability function P_B both consistent with $\mathcal{G}$ (see Eq. 1). Contrary to previous fragment-based or atom-based molecule-generation GFlowNet environments, where any backward action can lead to s_0 (removing nodes and edges sequentially will lead to an empty graph; Bengio et al., 2021), defining P_B in a reaction-based environment is non-trivial. This is because not every parent state (obtained by applying a reaction template backwards) will ensure that there exists a sequence of actions that leads all the way back to a building block, and therefore s_0. Note that the masking described in Section 2.1 is insufficient to account for this, as it does not ensure that the state obtained is *further* decomposable into building blocks. Consequently, to maintain a *pointed* DAG, no flow should be assigned to such a transition. A uniform backward policy, which is a standard choice in GFlowNet literature (Malkin et al., 2022), will fail at achieving this as it will assign positive flow to every backward action, including those leading to states that are not attainable from forward trajectories initialised in s_0 (see Figure 1-C). To address this issue, we explore a few training schemes for a parameterized P_B that force backward-constructed trajectories to end in s_0. We discuss the training schemes for P_B in Appendix B.4.

3 Results

3.1 Reaction-based MDP design

We compare GFlowNets using two action spaces – fragments vs. reactions – and report reward, diversity and synthesizability of the generated samples.

The results in Fig. 2 show that our MDP design achieves great improvement in terms of synthesisability, while preserving high rewards and diversity. We see that within the reaction environment, MDP design choices can further influence sample quality: a maximum trajectory length of 3 achieves better SA scores and AiZynthFinder (Genheden et al., 2020) retrosynthesis success (62%) compared to a maximum trajectory length of 4 (40%)[1]. Note that synthesisability metrics are correlated with

[1] The reactions available to AiZynthFinder are the US patent office (USPTO) set (Lowe), which are different to the reactants used by SynFlowNet, so one should not expect a 100% success score.

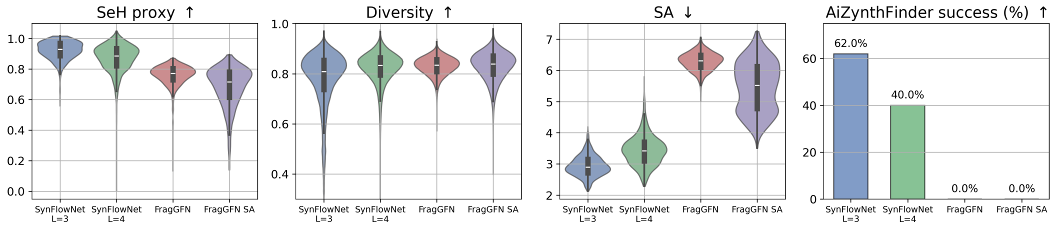

Figure 2: **Comparison across MDPs.** We evaluate four GFlowNet models: SynFlowNet is trained with an action space of chemical reactions and maximum trajectory lengths (L) of 3 and 4, FragGFN and FragGFN SA are trained with an action space of fragments, however the latter also optimises for synthetic accessibility (SA), besides the sEH binding proxy reward. SynFlowNet molecules are achieving higher binding scores and better synthesisability.

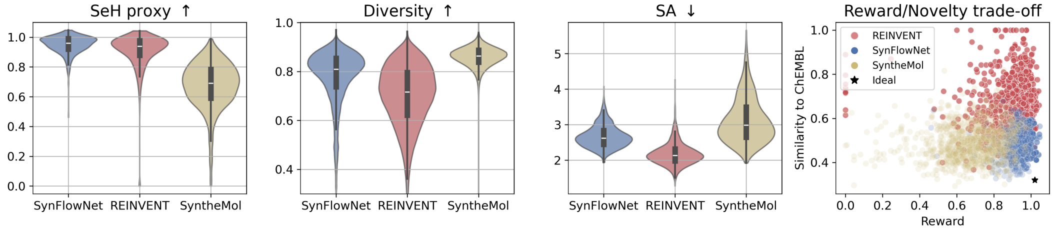

Figure 3: **SynFlowNet is competitive against other popular models from the literature.** SynFlowNet achieves a good balance between reward optimisation, diversity, synthesizability and novelty (assessed by maximum similarity to ChEMBL molecules). REINVENT stays close to its pretraining distribution, harming the novelty of the proposed molecules. SynFlowNet is closest to ideal.

molecule size (Skoraczyński et al., 2023), and that molecules assembled from longer synthesis routes are naturally larger (see Table A.2). Both FragGFN models score 0% with AiZynthFinder. Diversity, measured as average Tanimoto distances between molecular fingerprints, is preserved from a fragments to a reaction environment, albeit less so for shorter trajectories.

3.2 Further comparison to baselines

The comparison is summarized in Figure 3. SynFlowNet achieves comparable rewards to REINVENT, and better sample diversity. Although REINVENT does not use explicit knowledge of synthesizability in its generation process, it implicitly optimizes for it by being a likelihood model overfit on a curated ChEMBL dataset (Zdrazil et al., 2023a). This is reflected in our last sub-plot, where we look at the maximum similarity to ChEMBL molecules, as a proxy for novelty. This proxy is strong enough, as ChEMBL contains one of the largest collection of diverse drug-like molecules to date (Zdrazil et al., 2023b). While REINVENT seems to be competitive in terms of synthesizability and reward, it generates molecules with poor novelty, staying close to its pretraining distribution. Notably, SynFlowNet achieves high novelty for high-scoring molecules. Overall, SyntheMol scores comparably to SynFlowNet, however SynFlowNet achieves a better reward/diversity trade-off and SynFlowNet's capabilities are extended to explore larger state spaces (see Fig. C.1). We repeat the comparison on other targets, and again find that SynFlowNet outperforms SyntheMol (see App. C).

3.3 Improved MDP consistency through trained backward policy

We evaluate the effectiveness of our proposed training schemes for the backward policy P_B by measuring (1) whether P_B can ensure that backward-constructed trajectories reliably find a trajectory back to the initial state s_0 and (2) whether it brings any benefit to the forward policy. In Table A.3 we compare a fixed, uniform backward policy to three versions of a parameterized backward policy: a free policy, updated w.r.t. the trajectory balance loss, (see Eq. 1), a policy trained with maximum likelihood on the forward-generated trajectories and a policy that is allowed to explore the backward action space and trained with REINFORCE to find paths leading back to s_0. We observe that both the maximum likelihood and REINFORCE policies succeed in ensuring that backward flow is not lost outside of the MDP, as they manage to construct trajectories that start from terminal states sampled from P_F all the way to s_0. We refer to such routes as *solved routes (train)* in Table A.3, as the terminal states have been visited by the GFlowNet during training. We also test the ability of the trained policies to retrieve synthesis routes for molecules which have not been visited during training ($test$). Overall, the maximum likelihood and REINFORCE policies prove effective in guiding P_F to high reward modes, while ensuring MDP consistency.

3.4 Scaling to the entire Enamine set

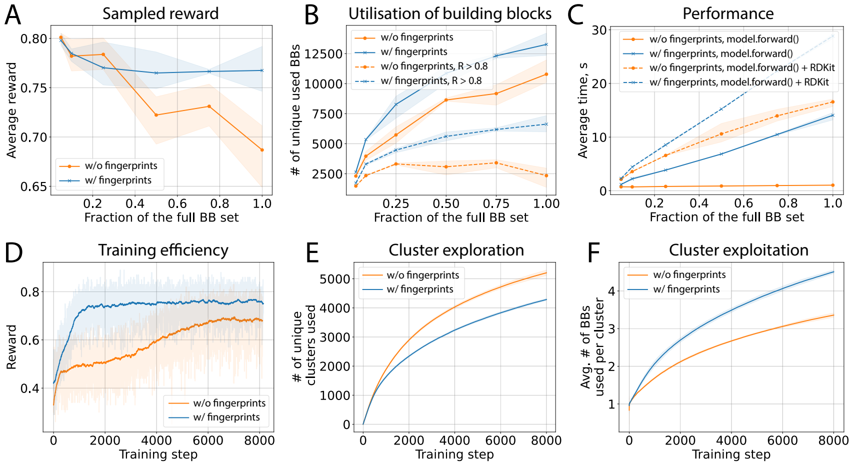

Figure 4: **Ablation over action-selection mechanism.** Average reward of samples (A), utilization of building blocks (B), and training times (C) for both models on different subsets of building blocks. Embeddings enable more efficient learning (D), as they help the agent navigate the set of available building blocks and tend to exploit more relevant BB clusters (F) rather than explore new clusters (E). For each experiment, three models trained with different random seeds were used.

In this section, we study the ability of SynFlowNet to scale to large sets of building blocks using the method proposed in Section C.1 as opposed to using a learnable embedding for each building block. As shown in Figure 4A, increasing the number of building blocks in general negatively affects the quality of the sampled molecules in terms of reward. However, the model with fingerprints consistently outperforms the former and makes the reward degradation effect almost negligible. Next, we study whether models are able to use the blocks they are given. As shown in Figure 4B, the model with fingerprints consistently uses more unique building blocks than the baseline model. Fingerprints enable the use of a 5-fold larger set of blocks for the same quality level, thus significantly surpassing the baseline samples in terms of diversity. As shown in Figure 4D, fingerprints enable more efficient learning on a large building block set. The former model tends to explore more clusters and less unique building blocks within clusters. The model with fingerprints, on the contrary, is able to focus on clusters that maximise the reward.

4 Conclusion

In this work, we discuss the application of GFlowNets to *de novo* molecular design paired with forward synthesis. We demonstrate that an action space of chemical reactions is an effective way of enforcing synthesisability, and that pairing it with GFlowNets excels in terms of diversity. Comparisons to state-of-the-art baselines emphasised that SynFlowNet explores novel regions of the chemical space. We also proposed a novel paradigm for training the backward policy in the GFlowNet and in doing so we improved and validated the correctness of our MDP design. Furthermore, we studied the building block exploration and exploitation mechanisms of SynFlowNet, showing efficient scaling to using hundreds of thousands of building blocks. Finally, as a proof of concept for the adaptability of SynFlowNet to real drug discovery programs, we showed that the framework can be specialised to target-specific molecule generation by making use of experimental fragment screens.

Acknowledgements

We thank the entire team at Valence Labs for participating in discussions and providing feedback on this work. We thank Enamine for providing the building blocks and for useful conversations. We thank Daniel Cutting, Austin Tripp and Jeff Guo for discussions regarding baselines. Miruna Cretu is supported by the SynTech CDT at the University of Cambridge and part of the work was completed during an internship at Valence Labs.

References

A Alhossary, SD Handoko, Y Mu, and CK Kwoh. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015. doi: 10.1093/bioinformatics/btv082.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. *arXiv preprint arXiv:2106.04399*, 2021.

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, 2020. doi: 10.1021/acs.jcim.0c00915. URL `https://doi.org/10.1021/acs.jcim.0c00915`. PMID: 33118816.

Melissa L Boby, Daren Fearon, Matteo Ferla, Mihajlo Filep, Lizbé Koekemoer, Matthew C Robinson, COVID Moonshot Consortium‡, John D Chodera, Alpha A Lee, Nir London, et al. Open science discovery of potent noncovalent sars-cov-2 main protease inhibitors. *Science*, 382(6671):eabo7201, 2023.

John Bradshaw, Brooks Paige, Matt J. Kusner, Marwin H. S. Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. *CoRR*, abs/1906.05221, 2019.

John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. Barking up the right tree: an approach to search over molecule synthesis dags. *Advances in neural information processing systems*, 33:6852–6866, 2020.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

Alexander Button, Daniel Merk, Jan A Hiss, and Schneider Gisbert. Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis. *Nature Machine Intelligence*, 1(7):307–315, 2019.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2022.

Tobiasz Cieplinski, Tomasz Danel, Sabina Podlewska, and Stanislaw Jastrzebski. Generative models should at least be able to design molecules that dock well: A new benchmark. *Journal of Chemical Information and Modeling*, 63(11):3238–3247, 2023. doi: 10.1021/acs.jcim.2c01355. URL `https://doi.org/10.1021/acs.jcim.2c01355`. PMID: 37224003.

E. J. Corey and X.-M. Cheng. *The Logic of Chemical Synthesis*. New York: John Wiley and Sons, 1989.

Yuanqi Du, Arian R Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Liò, Philippe Schwaller, and Tom L Blundell. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, pp. 1–16, 2024.

Enamine. Enamine. `https://enamine.net`. Accessed: 28.11.2023.

Wenhao Gao, Rocío Mercado, and Connor W. Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. *CoRR*, abs/2110.06389, 2021.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization, 2022. URL `https://arxiv.org/abs/2206.12411`.

S. Genheden, A. Thakkar, V. Chadimová, J. L. Reymond, O Engkvist, and Esben Bjerrum. Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning. *J Cheminform*, 12:70, 2020. doi: 10.1186/s13321-020-00472-1.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1263–1272, 2017.

Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, Sarath Chandar, and Yoshua Bengio. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3668–3679, 2020.

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. *arXiv preprint arXiv:1705.10843*, 2018.

Jeff Guo and Philippe Schwaller. Directly optimizing for synthesizability in generative molecular design using retrosynthesis models, 2024. URL https://arxiv.org/abs/2407.12186.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies, 2017. URL https://arxiv.org/abs/1702.08165.

Charles Harris, Kieran Didi, Arian Jamasb, Chaitanya Joshi, Simon Mathis, Pietro Lio, and Tom Blundell. Posecheck: Generative models for 3d structure-based drug design produce unrealistic poses. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023a.

Charles Harris, Kieran Didi, Arne Schneuing, Yuanqi Du, Arian Rokkum Jamasb, Michael M Bronstein, Bruno Correia, Pietro Lio, and Tom Leon Blundell. Flexible small-molecule design and optimization with equivariant diffusion models. In *ICLR 2023-Machine Learning for Drug Discovery workshop*, 2023b.

M Hartenfeller, H Zettl, M Walter, M Rupp, F Reisen, E Proschak, S Weggen, H Stark, and G. Schneider. DOGS: reaction-driven de novo design of bioactive. *PLOS Computational Biology*, 8(2), 2012. doi: 10.1371/journal.pcbi.1002380.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 8867–8887, 2022.

Julien Horwood and Emmanuel Noutahi. Molecular design in synthetically accessible chemical space via deep reinforcement learning. *ACS Omega*, 5(51):32984–32994, December 2020. doi: 10.1021/acsomega.0c04153.

Ilia Igashov, Hannes Stärk, Clément Vignac, Arne Schneuing, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, pp. 1–11, 2024.

John D. Imig and Bruce D. Hammock. Soluble epoxide hydrolase as a therapeutic target for cardiovascular diseases. *Nat Rev Drug Discov*, 8(2):794–805, 2009. doi: 10.1038/nrd2875.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective GFlowNets. *arXiv preprint arXiv:2210.12765*, 2023.

Hyosoon Jang, Yunhui Jang, Minsu Kim, Jinkyoo Park, and Sungsoo Ahn. Pessimistic backward policy for gflownets. *arXiv preprint arXiv:2405.16012*, 2024.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction Tree Variational Autoencoder for Molecular Graph Generation. *arXiv preprint arXiv:1802.04364*, 2019.

Vicky Jung, Kenneth Lopez Perez, Lexin Chen, Kate Huddleston, and Ramon Alain Miranda Quintana. Efficient clustering of large molecular libraries. *bioRxiv*, pp. 2024–08, 2024.

Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric Xing. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 3393–3403, 2020.

Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Cheng-Hao Liu, Mike Tyers, and Robert A Batey. Rgfn: Synthesizable molecular generation using gflownets. *arXiv preprint arXiv:2406.08506*, 2024.

Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10:1–24, 2018.

Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16, 2024.

D. Lowe. Chemical reactions from US patents. `https://figshare.com/articles/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873`. Accessed: 31.04.2018.

Shitong Luo, Wenhao Gao, Zuofan Wu, Jian Peng, Connor W. Coley, and Jianzhu Ma. Projecting molecules into synthesizable chemical spaces, 2024. URL `https://arxiv.org/abs/2406.04628`.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *CoRR*, abs/2201.13259, 2022.

AD Mesecar et al. A taxonomically-driven approach to development of potent, broad-spectrum inhibitors of coronavirus main protease including sars-cov-2 (covid-19). *Be Publ*, 10, 2020.

Sobhan Mohammadpour, Emmanuel Bengio, Emma Frejinger, and Pierre-Luc Bacon. Maximum entropy gflownets with soft q-learning, 2024. URL `https://arxiv.org/abs/2312.14331`.

Christopher W Murray and David C Rees. The rise of fragment-based drug discovery. *Nature chemistry*, 1(3):187–192, 2009.

Christopher W Murray, Maria G Carr, Owen Callaghan, Gianni Chessari, Miles Congreve, Suzanna Cowan, Joseph E Coyle, Robert Downham, Eva Figueroa, Martyn Frederickson, et al. Fragment-based drug discovery applied to hsp90. discovery of two lead series with high ligand efficiency. *Journal of medicinal chemistry*, 53(16):5942–5955, 2010.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017a.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(48), 2017b. doi: 10.1186/s13321-017-0235-x.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t. URL `https://doi.org/10.1021/ci100050t`. PMID: 20426451.

Julien Roy, Pierre-Luc Bacon, Christopher Pal, and Emmanuel Bengio. Goal-conditioned gflownets for controllable multi-objective molecular design. *arXiv preprint arXiv:2306.04620*, 2023.

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. *arXiv preprint arXiv:2210.13695*, 2023.

Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1): 120–131, 2017. doi: 10.1021/acscentsci.7b00512.

Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

Tony Shen, Mohit Pandey, Jason Smith, Artem Cherkasov, and Martin Ester. Tacogfn: Target conditioned gflownet for structure-based drug design. *arXiv preprint arXiv:2310.03223*, 2023.

Grzegorz Skoraczyński, Mateusz Kitlas, Błażej Miasojedow, and Anna Gambin. Critical assessment of synthetic accessibility scores in computer-assisted synthesis planning. *Journal of Cheminformatics*, 15, 2023.

Megan Stanley and Marwin Segler. Fake it until you make it? generative de novo design and virtual screening of synthesizable molecules. *Current Opinion in Structural Biology*, 82:102658, 2023.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Kyle Swanson, Gary Liu, Denise B Catacutan, Autumn Arnold, James Zou, and Jonathan M Stokes. Generative ai for designing and validating easily synthesizable and structurally novel antibiotics. *Nature Machine Intelligence*, 6(3):338–353, 2024.

Shidi Tang, Ji Ding, Xiangyu Zhu, Zheng Wang, Haitao Zhao, and Jiansheng Wu. Vina-gpu 2.1: towards further optimizing docking speed and precision of autodock vina and its derivatives. *bioRxiv*, 2023. doi: 10.1101/2023.11.04.565429. URL https://www.biorxiv.org/content/early/2023/11/05/2023.11.04.565429.

Sherine E Thomas, Patrick Collins, Rory Hennell James, Vitor Mendes, Sitthivut Charoensutthivarakul, Chris Radoux, Chris Abell, Anthony G Coyne, R Andres Floto, Frank Von Delft, et al. Structure-guided fragment-based drug discovery at the synchrotron: screening binding sites and correlations with hotspot mapping. *Philosophical Transactions of the Royal Society A*, 377(2147): 20180422, 2019.

Jos Torge, Charles Harris, Simon V Mathis, and Pietro Lio. Diffhopp: A graph diffusion model for novel drug design via scaffold hopping. *arXiv preprint arXiv:2308.07416*, 2023.

Oleg Trott and Arthur J. Olson. Autodock Vina: improving the speed and accuracy of docking with a new scoring func- tion, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010. doi: 10.1002/jcc.21334.

H. Maarten Vinkers, Marc R. de Jonge, Frederik F. D. Daeyaert, Jan Heeres, Lucien M. H. Koymans, Joop H. van Lenthe, Paul J. Lewi, Henk Timmerman, Koen Van Aken, and Paul A. J. Janssen. Synopsis: SYNthesize and OPtimize System in Silico. *Journal of Medicinal Chemistry*, 46(13): 2765–2773, 2003. doi: 10.1021/jm030809x.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

Andrew J Woodhead, Hayley Angove, Maria G Carr, Gianni Chessari, Miles Congreve, Joseph E Coyle, Jose Cosme, Brent Graham, Philip J Day, Robert Downham, et al. Discovery of (2, 4-dihydroxy-5-isopropylphenyl)-[5-(4-methylpiperazin-1-ylmethyl)-1, 3-dihydroisoindol-2-yl] methanone (at13387), a novel inhibitor of the molecular chaperone hsp90 by fragment based drug design. *Journal of medicinal chemistry*, 53(16):5956–5969, 2010.

Moksh Jain Yoshua Bengio, Kolya Malkin. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. https://github.com/recursionpharma/gflownet, 2024. Accessed: 14.09.2024.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Chengxi Zang and Fei Wang. Moflow: An invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 617–626, 2020. doi: 10.1145/3394486.3403104.

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, 2023a. doi: 10.1093/nar/gkad1004.

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 2023b.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

A Background and Related Work

A.1 GFlowNets

GFlowNets (Bengio et al., 2021) are a class of probabilistic models that learn a stochastic policy to generate objects x through a sequence of actions, with probability proportional to a reward $R(x)$. The sequential construction of objects x can be described as a trajectory $\tau \in \mathcal{T}$ in a directed acyclic graph (DAG) $G = (S, \mathcal{E})$, starting from an initial state s_0 and using actions a to transition from a state to the next: $s \rightarrow s'$. A GFlowNet uses a forward policy $P_F(-|s)$, which is a distribution over the children of state s, to sample a sequence of actions based on the current states. Similarly, a backward policy $P_B(-|s)$ is the distribution over the parents of state s, and can be used to calculate probabilities of backward actions, leading from terminal to initial states. The training objective which we adopt in this paper is trajectory balance (Malkin et al., 2022):

$$\mathcal{L}_{\text{TB}}(\tau) = \left(\log \frac{Z_\theta \prod_{t=1}^{T} P_F(s_{t+1}|s_t;\theta)}{R(x) \prod_{t=1}^{T} P_B(s_{t-1}|s_t;\theta)} \right)^2 \tag{1}$$

used to learn the forward and backward policies $P_F(-|s;\theta)$ and $P_B(-|s;\theta)$ parameterized by θ and to estimate the partition function $Z_\theta \approx F(s_0) = \sum_{\tau \in \mathcal{T}} F(\tau)$.

Bengio et al. (2021) have used GFlowNets to generate molecules with high binding affinity to a protein target by linking fragments to form a junction tree (Jin et al., 2019). For generative chemistry, the framework was extended to multi-objective optimisation (Jain et al., 2023; Roy et al., 2023), where the model was trained to simultaneously optimise for binding affinity to the protein target, Synthetic Accessibility (SA), drug likeness (QED) and molecular weight. Recently, Shen et al. (2023) extended the framework to pocket-conditioned molecular generation, employing the same action space to generate molecules conditioned on different protein targets.

A.2 GFlowNets with a trained backward policy

GFlowNets train a forward policy P_F to match the backward policy P_B according to Eq. 1. The choice of P_B therefore impacts the training of GFlowNets and sample quality. Despite this relationship, the choice of backward policy in GFlowNets has attracted limited attention but for a few works discussed here. Malkin et al. (2022) proposed parameterizing the backward policy and training P_B and P_F simultaneously using the trajectory balance objective (Eq. 1). They also proposed fixing the backward policy to a uniform distribution when modeling the distribution over parents states proves difficult. Mohammadpour et al. (2024) compare a maximum-entropy GFlowNet to GFlowNets with a uniform backward policy. Closer to our work, Jang et al. (2024) propose Pessimistic GFlowNets, which use *maximum likelihood* over observed trajectories to train P_B. This ensures that the flow induced by the backward policy is concentrated around observed (training) states, making the model pessimistic about unobserved intermediate states having flow. In this work, we address the idea of ensuring that backward-generated trajectories belong to the GFlowNet MDP. In doing so, we propose a data-driven solution to a MDP design challenge.

A.3 Generative models for molecule design

A large number of works have been proposed for generative molecular design (Du et al., 2024). Early methods employed techniques such as variational autoencoders (VAEs) (Gómez-Bombarelli et al., 2018), deep reinforcement learning (Segler et al., 2017; Olivecrona et al., 2017b), and generative adversarial networks (GANs) (Guimaraes et al., 2018; Cao & Kipf, 2022). More recently, methods like normalizing flows (Zang & Wang, 2020) and diffusion models (Hoogeboom et al., 2022; Schneuing et al., 2023; Harris et al., 2023b; Torge et al., 2023; Igashov et al., 2024) have gained significant attention due to their ability to model complex distributions more effectively. Despite these advances, challenges remain, particularly in ensuring that generated molecules adhere to physical and chemical constraints (Stanley & Segler, 2023; Harris et al., 2023a).

A.4 Synthesis-aware molecule generation

The idea of tackling molecule generation and synthesis simultaneously has been investigated early by Vinkers et al. (2003), who introduced SYNOPSIS, which generates molecules from a starting dataset of available compounds, relying on applying chemical modifications to functional groups and assessing the value of the product with a fitness function. The works of Bradshaw et al. (2019) and Korovina et al. (2020) provided early neural models for one-step synthetic pathways. Bradshaw et al. (2020) generalized this idea as a generative model for synthesis DAGs, which can be optimized in a VAE or RL setup. Gottipati et al. (2020) used reinforcement learning to generate compounds from reactions and commercially available reactants. Gao et al. (2021) formulate an MDP to model the generation of synthesis trees, which can be optimized with respect to the desired properties of a product molecule. Luo et al. (2024) proposed a model that can project unsynthesizable molecules from existing generative models to synthesizable chemical space by utilizing postfix notations to represent synthesis pathways. Guo & Schwaller (2024) showed that retrosynthesis models can be treated as an oracle in goal-directed molecule generation. Concurrently to our work, Koziarski et al. (2024) propose a similar GFlowNet-based framework for synthesizable molecular generation but using a different set of reaction templates and making use of a different backward policy.

While our work resembles an RL setting for synthesis-aware molecular generation (Gottipati et al., 2020; Horwood & Noutahi, 2020), the key difference lies in the sampling distribution of the learned model. Contrary to RL, the GFlowNet objective is not to generate the single highest-return sequence of actions, but rather to maximise both performance and diversity by sampling terminal states proportionally to their reward. This is especially useful in the context of molecule generation, where we want to explore different modes of the distribution of interest.

B Extended methods

B.1 Reward functions

We train SynFlowNet for a number of reward functions/targets. In some cases (e.g. when benchmarking against the Fragment-based GFlowNet), we also perform multiple objective optimisation with SA score.

sEH Our main reward function is defined as the normalized negative binding energy as predicted by a pretrained proxy model, available from Bengio et al. (2021) and trained on molecules docked with AutoDockVina (Trott & Olson, 2010) for the sEH (soluble epoxide hydrolase) protein target, a well studied protein which plays part in respiratory and heart disease (Imig & Hammock, 2009). The proxy model, which utilizes the weights from Bengio et al. (2021), was trained using a message-passing neural network (MPNN) (Gilmer et al., 2017) that processes atom graphs as input. Details of the model architecture are available in Bengio et al. (2021). It was trained on a dataset of 300,000 randomly generated molecules, achieving a test mean squared error (MSE) of 0.6. Note that the reward scale in our results differs from the original GFlowNet publication, with rewards adjusted by a factor of 1/8 in our analysis.

gsk3β and DRD2 We also employ two oracle functions from the PMO Gao et al. (2022) benchmark, which provide machine learning proxies trained fit to experimental data to predict the bioactivities against their corresponding disease targets. The two targets we use here are gsk3β Li et al. (2018) and dopamine receptor D2 (DRD2) (Olivecrona et al., 2017a).

Easy adoption to other targets using GPU-accelerated Vina docking Finally, we wish for users to rapidly be able to adapt SynFlowNet to learn binding for their target of interest without having to retrain a new proxy or relying on slow docking simulations. We accomplish this using the new GPU-accelerated Vina-GPU 2.1 docking algorithm (Tang et al., 2023; Alhossary et al., 2015). For our experiment, we use the `PDB:6W63` (Mesecar et al., 2020) structure for the SARS-CoV-2 main protease (Mpro) target and `PDB:2XJX` (Murray et al., 2010) for Heat Shock Protein 90 (HSP90) target. Receptors are prepared for docking with `prepare_receptor4.py` and the center of the docking is defined as the center of mass for the ligand with a size of 25 Å in accordance with previous work (Buttenschoen et al., 2024).

B.2 Chemical reactions

Masking Prior to sampling actions in both forward and backward direction, we ensure that the reactions and building blocks to be sampled are compatible with the current state using masks,

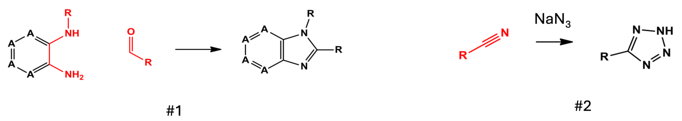

Figure A.1: **Example of reaction templates** (Hartenfeller et al., 2012). The templates act as rules, which match any molecule that left side (before the arrow) as a subgraph. The matching part is then transformed into the right hand side of the rule. “R” represents any group, “A” represents an aromatic atom. Note the implicit reagent used in reaction #2.

obtained by checking for substructure match between the reaction template and the reactant (for forward actions) or product (for backward actions) molecules. We also enforce through masking that at least one of the resulting reactants when running a backward reaction is a building block.

Data We use commercially available building blocks (BBs) from Enamine, which are small fragments of molecules prepared in bulk to be readily synthesised into candidate molecules. Reaction templates are obtained from two publicly available template libraries (Button et al., 2019; Hartenfeller et al., 2012). After preprocessing, we obtain a total of 105 templates: 13 uni- and 92 bi-molecular reactions respectively (see App. B.2). We also use 12 Enamine REAL reactions in a small number of experiments (see Section C.4 and App. B.2).

Reaction pre-processing and action masking Reaction templates pre-processing was necessary to ensure that the templates could be run backwards. All templates containing wildcards (*) in the reactant SMARTS were duplicated with replacements for all atoms that the wildcard substituted for.

When running experiments with a REINFORCE backward policy (which requires to sample trajectories backward on-policy), we enforced additional masking that forward reactions are sampled if and only if they are reversible (i.e. when applying the template backward on the product, the same reactants are obtained as the ones utilised by the template in the forward direction).

Note that reaction templates can be *uni-molecular* (employing one reactant, Fig. A.1 #2) or *bi-molecular* (employing two reactants, Fig. A.1 #1).

Enamine REAL Space Reactions For the case study in Section C.4, we used reactions from Enamine to stay close to an experimental setup where the molecules generated would be readily purchasable from Enamine. Enamine assembled a molecular space called the REAL space, which is a vast catalogue of 48B purchasable compounds[2]. We use a subset of Enamine reactions available from Swanson et al. (2024) which produces 93.9% of the REAL space[3].

B.3 Model and training

A graph neural network based on a graph transformer architecture (Yun et al., 2019) is used to parameterize the forward and backward policies. The model's action space is defined using separate MLPs for each action type (see Figure 1). Our model is trained in an online fashion, meaning that it learns exclusively from trajectories sampled from the GFlowNet policy, without relying on an external dataset of trajectories or a set of target molecules. Note however that this framework is compatible with offline training, which makes use of such datasets as starting point for exploring the molecular space.

SynFlowNet Training We adapt the framework from Bengio et al. (2021) to train a GFlowNet sampler over a space of synthesisable molecules, which are assembled from an action space of chemical reactions and reactants. A graph neural network with a graph transformer architecture (Yun et al., 2019) is used to produce a state-conditional distribution over the actions. A state is represented as a molecular graph in which nodes contain atom features. Edge attributes are bond type and the indices of the atoms which are its attachment points. This representation is augmented with a fully-connected virtual node, which is an embedding of the conditional encoding of the

[2] https://enamine.net

[3] Note: this dataset contains a single tri-molecular reaction which has been removed here for simplicity

desired sampling temperature, obtained using an MLP. The sampling temperature is controlled by a temperature parameter β, which also plays a role in reward modulation, allowing for exponential scaling of the rewards (by making rewards received during training equal to R^{β}). We experimented with sampling β from multiple distributions, and use a constant distribution in the reported results in this paper. We used a thermometer encoding of the temperature (Buckman et al., 2018).

The model is trained using the trajectory balance objective (Malkin et al., 2022) and thus is parameterized by forward and backward action distributions P_F and P_B and an estimation of the partition function $Z = \sum_{\tau \in \mathcal{T}} F(\tau)$.

For the state space estimation experiment, we used $R = 1$, $10\,000$ training steps and varying trajectory lengths. For the rest of the SynFlowNet experiments with SeH proxy as reward, we used $\beta = 32$, 5000 training steps and varying trajectory lengths (see main text). For training with the DRD2 and gsk3β targets as reward, we use a maximum trajectory length of 4 and a reward exponent $\beta = 16$. For the backward policy training, see App. B.4. The rest of the hyperparameters are fixed and are presented in Table A.1.

Unless otherwise specified, all experiments below use a backward policy trained with maximum likelihood, and Morgan fingerprint embeddings for our action space, with a library of 10,000 Enamine building blocks and 105 reaction templates.

FragGFN training We obtained fragments and their attachment points from Enamine building blocks by following the protocol provided by Jin et al. (Jin et al., 2019). The model trained with the sEH reward was trained with the default implementation in Yoshua Bengio (2024), and an optimised reward exponent $\beta = 64$. The model optimising for synthetic accessibility, as well as sEH binding, was trained with a reward obtained by multiplying the two scores.

Soft Q-learning training We implemented a version of Soft Q-learning (SQL) (Haarnoja et al., 2017), an energy-based policy learning method, that operates on SynFlowNet's MDP. We optimised the training procedure by performing both manual and grid searches across several values of entropy regularisation parameter α and reward scaling parameter β. For β, contrarily to GFlowNets, since the method does not only learn a policy but tries to estimate the Q-values of each actions directly, we found that using large values of reward scaling such as $\beta = 64$, which are common for GFlowNets, would destabilise the algorithm and had to be lowered to $\beta = 4$ or $\beta = 2$. For α, we tried several values that would strike the best tradeoff between allowing the model to find high-performing molecules while maximising diversity and avoiding to collapse the agent's distribution on only a few modes, with our best model using $\alpha = 0.01$.

REINVENT training We benchmarked our approach against REINVENT4 (Loeffler et al., 2024)[4]. Following the setup described in their methodology, we fine-tuned the REINVENT prior model using reinforcement learning. We change little else other than the reward function used to train the model. The training was conducted with the default batch size of 100 over 3,000 training steps, resulting in a total of 300,000 oracle calls, which is consistent with the number of oracle calls used in SynFlowNet experiments.

SyntheMol generation We run SyntheMol's (Swanson et al., 2024) Monte Carlo Tree Search (MCTS) algorithm using their standard sets of $139\,493$ building blocks and 13 chemical reactions from the Enamine REAL space, but replace the provided bioactivity prediction models with the reward functions used in this study (Sec. B.1). We use the publicly available code repository[5] and perform $50\,000$ rollouts. Other hyperparameters were kept at default settings, including a maximum of 1 reaction and exploration parameter $c = 10$. Before the search, building block scores were pre-computed using the target reward functions. For the final selection, we sample 1000 molecules randomly from all returned molecules.

[4] We used the code available at `https://github.com/MolecularAI/REINVENT4`

[5] `https://github.com/swansonk14/SyntheMol`

Hyperparameters	Values
Batch size	64
Number of GNN layers	4
GNN node embedding size	128
Graph transformer heads	2
Learning rate (P_F)	10^{-4}
Learning rate (P_B)	10^{-4}
Learning rate (Z)	10^{-3}
max_nodes	9

Table A.1: Hyperparameters used in our SynFlowNet training pipelines.

B.4 Backward Policy Training Algorithms

Training the backward policy We first explore a training scheme for P_B which makes use of forward-generated trajectories. Similarly to Jang et al. (2024), we train P_B using the maximum likelihood objective over trajectories generated from P_F:

$$\mathcal{L}_B(\theta_B) = \mathbb{E}_{\tau \sim P_F}[-\log P_B(\tau; \theta_B)]. \tag{2}$$

In that setting, we (1) generate trajectories using P_F, (2) update P_F according to the trajectory balance objective in Equation 1 and (3) update P_B using these same trajectories according to Equation 2 (see see Algorithm 1).

While the maximum-likelihood approach presented above is sufficient to limit P_B in allocating flow to paths that do not connect back to s_0, it restricts the model's exploration by encouraging P_F to collapse on a single path for each terminal molecule. To allow P_B to ban erroneous paths while retaining a higher entropy, we also explore the use of policy gradient methods. Specifically, we explore maximising an expected *backwards* reward via REINFORCE (Williams, 1992), which is suitable for short trajectory environments like our reaction-based MDP:

$$J_B(\theta_B) = E_{\tau \sim P_F, P_B}[R_B(\tau)] - \alpha H(P_B). \tag{3}$$

Here, $H(P_B) = -\mathbb{E}_{\tau \sim P_B}[\log(P_B(\tau))]$ is the entropy term and the reward R_B is set to 1 for a trajectory that ends in s_0 and -1 otherwise. In this setting we train the backward policy not only on the trajectories generated by the forward policy, but also on newly generated backward trajectories sampled directly from P_B (see Algorithm 2).

For the maximum likelihood backward policy, at each training iteration, P_B is updated to minimize the maximum likelihood loss over trajectories generated in that batch from P_F. Similarly for a REINFORCE P_B, we train P_F as above, but maintain a replay buffer. To train P_B, we sample terminal states from the buffer and sample trajectories backwards from P_B, which are used in a REINFORCE update. To improve training stability, we also used trajectories generated from P_F (in a 1:1 ratio) to update P_B.

Contrary to the rest of the models (see App. B.3), the backward policy models were trained for 8000 steps, with reward exponent $\beta = 64$ and max_len $= 5$. The results in Section 3.3 are reported for a REINFORCE loss with an entropy multiplier term $\alpha = 1.0$.

Algorithm 1 Training of Maximum Likelihood Backward Policy for GFlowNets

```
Initialize the forward policy $P_F$, backward policy $P_B$, and $Z_\theta$.
repeat
    Sample a batch of trajectories $\{\tau^{(n)}\}_{n=1}^N$ from $P_F$.
    Update $P_F$ and $Z_\theta$ to minimize $\mathcal{L}_{TB}$ using $\{\tau^{(n)}\}_{n=1}^N$.
    Update $P_B$ to minimize $\mathcal{L}_B$ over $\{\tau^{(n)}\}_{n=1}^N$.
until convergence
```

Interestingly, training the backward policy to navigate back to s_0 is analogous to the retrosynthesis problem (Corey & Cheng, 1989; Segler et al., 2018). We employ this approach of training P_B against a different objective than P_F to palliate to a design challenge of the MDP, but a similar strategy could also be employed to fold additional preferences over different synthesis routes leading to the same

Algorithm 2 Training of REINFORCE Backward Policy for GFlowNets

```
Initialize the replay buffer $\mathcal{B}$, forward policy $P_F$, backward policy $P_B$, and $Z_\theta$.
repeat
    Sample a batch of trajectories $\{\tau_F^{(n)}\}_{n=1}^N$ from $P_F$.
    Update $\mathcal{B} \leftarrow \mathcal{B} \cup \{\tau_F^{(k)}\}_{n=1}^N$.
    Sample $k$ random trajectories from $\mathcal{B}$ and extract their final states $s_f$ to sample backward trajectories $\{\tau_B\}_{k=1}^K$ from $P_B$.
    Update $P_F$ and $Z_\theta$ to minimize $\mathcal{L}_{TB}$ using $\{\tau_F^{(n)}\}_{n=1}^N$.
    Update $P_B$ to minimize $J_B$ over $\{\tau_B^{(k)}\} \cup \ \{\tau_F^{(n)}\}$.
until convergence
```

terminal state into the system, for example to take into account the synthesis costs of a particular path. This possibility is left to future work.

B.5 SynFlowNet scaling

For Section 3.4, to compute clusters of building blocks, we used BitBIRCH (Jung et al., 2024), a recent adaptation of Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) algorithm (Zhang et al., 1996), recently proposed for efficient clustering of large molecular libraries.

C Extended Results

C.1 Scaling of the reactant space to large number of building blocks

We leverage the properties of GFlowNets (Bengio et al., 2021) to estimate the size of the state space induced by our action space. Specifically, noting that GFlowNets learn $\log Z = \log \sum_x R(x)$, we train a model with $R = 1$ for all terminal states to estimate their total count. We do so for different numbers of building blocks and different maximum trajectory lengths (L), and find that SynFlowNet using $L = 3$ and 10k building blocks matches the size of the Enamine REAL space (Enamine), and that the size of the space quickly increases with the number of building blocks. We use our full set of 105 reactions. Note how reaction-constrained models considerably limit the exploration of the chemical space, with a fragments GFlowNet exploring a space ∼10 orders of magnitude larger.

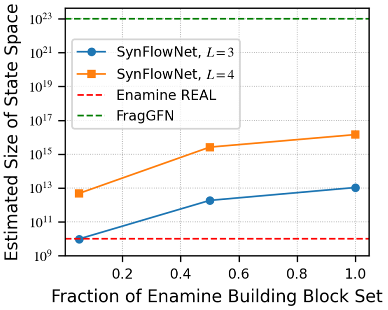

Figure A.2: **Estimated size of state spaces.** The full building block set contains 221,181 molecules. L is the maximum trajectory length in the GFlowNet.

To ensure synthetic accessibility of all samples, our model is inherently constrained by the initial set of available building blocks (BBs). To cover a large chemical space, it is thus crucial to use an extensive BB collection. The model must demonstrate scalability to accommodate larger sets of BBs, both in terms of training efficiency and overall performance. To do so, we change the representation of the BBs and their selection mechanism, as shown in Figure 1B. Instead of the weight matrix of the mapping from hidden units to logits associated with BBs to be randomly initialized, it is *fixed* to be the matrix of binary Morgan fingerprints (Rogers & Hahn, 2010).

C.2 GFlowNets as samplers of chemical space

Our results support a number of claims: (i) a reaction-based MDP greatly improves the synthesisability of generated molecules (Sec. 3.1), (ii) employing GFlowNets enables much more diverse molecule sampling over RL (Sec. 3.4), (iii) learning a backward policy results in higher-reward molecules and enables finding retrosynthetic pathways for molecules belonging to our state-space (Sec. 3.3), and (iv) our method of sampling molecules based on chemical fingerprints allows for efficient scaling to large chemical spaces (Sec. 3.4). Finally, we show that this approach also allows to improve results for particular programs by curating the set of building blocks based on target-specific experimental data (Sec C.4). Unless otherwise specified, all experiments below use a backward policy trained with maximum likelihood, and Morgan fingerprint embeddings for our action space, with a library of 10,000 Enamine building blocks and 105 reaction templates.

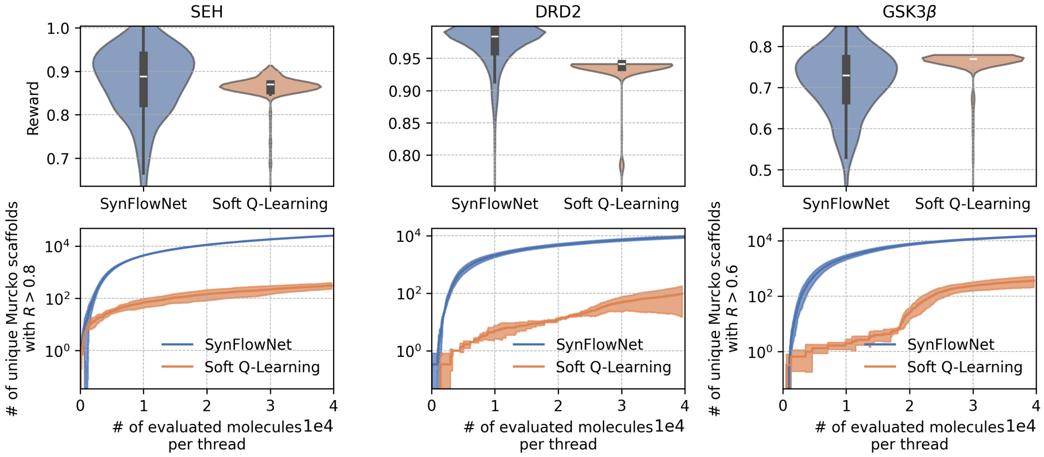

Figure A.3: **Comparison between GFlowNet and RL.** SynFlowNet (GFlowNet with synthesis actions) discovers more modes compared to entropy-regularised RL trained on the same state & action space.

While the works of Gottipati et al. (2020) and Horwood & Noutahi (2020) are closest to our method, where RL is paired with a synthesis action space, their code is not publicly available. We reproduce their setting by pairing an RL algorithm with our MDP. We train our model with a soft Q-learning (Haarnoja et al., 2017) objective and compare to SynFlowNet on three different reward functions: sEH, GSK3β and DRD2.

In Fig. A.3, we show the reward distribution of generated samples after equal numbers of training steps. We notice that soft Q-learning collapses to sampling a narrower distribution of high-reward molecules for all targets. This is also shown in the number of Bemis-Murcko scaffolds counted for molecules with rewards above a certain threshold, where soft Q-learning quickly collapses (note the logarithmic scale of the y-axis).

C.3 Comparison to baselines

We retrain the GFlowNet model proposed by Bengio et al. (2021) with fragments derived from our building blocks set. This ensures that we use similar chemical spaces and that we get as close as possible to a fair comparison between the two MDPs. In the fragment space, it is common practice to optimise for synthetic accessibility scores to improve synthesizability from the model. We therefore train two versions of the FragGFN model: one using the sEH binding proxy as reward function, and another that optimizes for both sEH binding and synthetic accessibility (SA) score. SynFlowNet was only trained with the sEH proxy as reward.v

We compare SynFlowNet to strong baselines from the literature. First, REINVENT uses a policy-gradient method to tune an RNN pre-trained to generate SMILES strings (Blaschke et al., 2020; Loeffler et al., 2024). REINVENT has been shown to outperform many models in terms of sample efficiency (Gao et al., 2022) and proposing realistic 3D molecules upon docking (Cieplinski et al., 2023). Second, SyntheMol shares a similar MDP design with SynFlowNet, but is a search-based method using Monte Carlo Tree Search (Swanson et al., 2024). Contrary to our model, synthesis pathways in SyntheMol contain one reaction, and its estimated state space has 30 billion molecules, reached from the Enamine building block set and 13 reactions (Swanson et al., 2024).

Table A.2 contains metrics for all the models and baselines we have run. Figure A.4 contains results for molecules generated with SyntheMol for the DRD2 target, compared to SynFlowNet. SynFlowNet is superior in finding high-reward molecules. We tried various hyperparameters for SyntheMol to optimise for the gsk3β target and were unsuccessful – no nonzero rewards were found.

Method	sEH proxy (↑)	Diversity (↑)	SA (↓)	AiZynth (↑)	QED (↑)	Mol. weight (↓)	ChEMBL Similarity (↓)
REINVENT	0.91 ± 0.01	0.68 ± 0.02	2.19 ± 0.02	0.95	0.57 ± 0.04	429.78 ± 23.02	0.64 ± 0.02
FragGFN	0.77 ± 0.01	0.83 ± 0.01	6.28 ± 0.02	0.00	0.30 ± 0.01	724.62 ± 32.39	0.24 ± 0.09
FragGFN SA	0.70 ± 0.01	0.83 ± 0.01	5.45 ± 0.05	0.00	0.29 ± 0.01	683.31 ± 59.92	0.21 ± 0.01
SyntheMol	0.64 ± 0.01	0.86 ± 0.01	3.08 ± 0.01	0.82	0.63 ± 0.01	412.24 ± 0.98	0.49 ± 0.01
Soft Q-Learning	0.80 ± 0.07	0.42 ± 0.04	2.63 ± 0.39	0.96	0.39 ± 0.02	408.02 ± 12.76	0.52 ± 0.02
SynFlowNet ($L = 3$)	0.92 ± 0.01	0.79 ± 0.01	2.92 ± 0.10	0.65	0.59 ± 0.02	365.23 ± 2.42	0.43 ± 0.01
SynFlowNet SA ($L = 3$)	0.94 ± 0.01	0.75 ± 0.02	2.67 ± 0.03	0.93	0.68 ± 0.01	358.27 ± 3.52	0.48 ± 0.01
SynFlowNet ($L = 4$)	0.88 ± 0.01	0.82 ± 0.01	3.54 ± 0.03	0.40	0.27 ± 0.01	557.49 ± 8.60	0.38 ± 0.01
SynFlowNet QED	0.86 ± 0.03	0.81 ± 0.03	4.02 ± 0.26	0.55	0.74 ± 0.04	398.50 ± 8.84	0.38 ± 0.01

Table A.2: **Comparison to baselines.** Results obtained by averaging over 1000 random molecules sampled from the trained models. Standard errors obtained from training using 3 seeds. Due to high computational cost, AiZynthFinder scores are computed over 100 random samples.

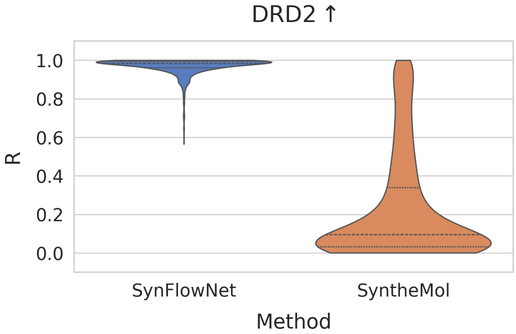

Figure A.4: Comparison between SynFlowNet and SyntheMol with DRD2 as reward. SyntheMol struggles to find high-reward molecules. We also run SyntheMol with GSK3β and were not able to optimise it. Results for SynFlowNet with both DRD2 and GSK3β are reported in Figure A.3.

C.3.1 Improved MDP consistency through trained backward policy

P_B policy	% of solved routes (train)	% of solved routes (test)	# of high reward modes from P_F
Uniform	$11.0 \pm 3.7\%$	$11.0 \pm 4.1\%$	$47,515.0 \pm 11,264.7$
Free	$67.3 \pm 3.7\%$	$1.0 \pm 0.8\%$	$6,754.3 \pm 4,980.1$
MaxLikelihood	$99.3 \pm 0.5\%$	$32.3 \pm 7.3\%$	$37,708.2 \pm 13,992.6$
REINFORCE	$100.0 \pm 0.0\%$	$44.3 \pm 2.6\%$	$55,387.6 \pm 28,886.3$

Table A.3: **Effect of different training paradigms for $\boldsymbol{P_B}$.** Training the GFN P_B ensures that backward-constructed trajectories belong to our MDP and can marginally improve P_F. % of solved routes refers to the ability of P_B to retrieve synthesis routes for on-policy (train) and off-policy (test) molecules reachable through our state space. Test molecules have not been seen during training. # of high reward ($R > 0.9$) modes from P_F is reported out of $\sim 500,000$ samples seen during training.

C.4 Guiding SynFlowNet with experimental fragment screens

One advantage of our framework is that building block sets can be used to specialize the model for a particular target. Fragment-based Drug Discovery (FBDD) (Thomas et al., 2019) is a major strategy to increase the efficiency of drug discovery campaigns. In FBDD, instead of screening large chemical spaces, a low number of small compounds (called fragments) are screened and experimentally validated to bind, and are then linked *in silico* to make full molecules.

We hypothesise that experimental data from x-ray fragment screens (Murray & Rees, 2009) can guide and enhance SynFlowNet's capabilities. As a proof-of-concept, we focus on the SARS-CoV-2 main protease (Mpro), leveraging strucural data from the COVID Moonshot project (Boby et al., 2023). We compile a building block set from Enamine with high similarity to fragments confirmed by x-ray crystallography (see Fig. A.6A). We find that biasing the building block library towards fragments with known protein-ligand complementarity increases the reward over randomly selected molecules (see Fig. A.6B). Further details on methods and results are given in App. Section C.6.

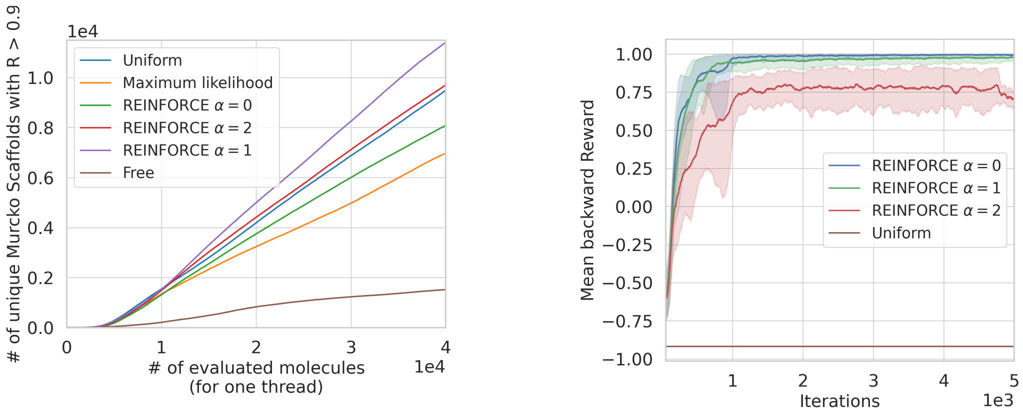

Figure A.5: (LHS) Effects of different parameterizations of the backward policy. For REINFORCE, (α) represents the multiplier to the entropy term. (RHS) REINFORCE is trained on-policy and the mean rewards of the sampled backward trajectories are plotted. Note that the backward reward is 1 if the backward-constructed trajectory ends in s_0 and -1 otherwise. A baseline for Uniform policy is shown, obtained by sampling 100 random trajectories backward from terminal objects in the DAG. Results averaged over 8 seeds.

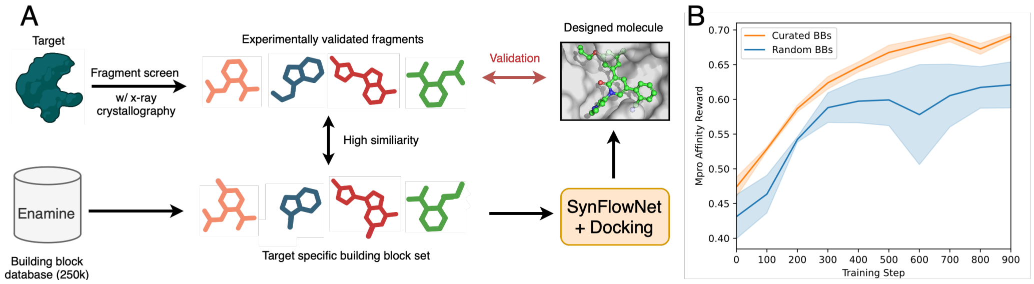

Figure A.6: **Effect of curating the building block set.** (A) We adapt SynFlowNet's building block library based on experimentally validated fragments for a given target. (B) The curated set improves Mpro rewards over random building blocks.

C.5 Example Trajectories

Figure A.7 shows examples of molecules and synthesis pathways generated from SynFlowNet.

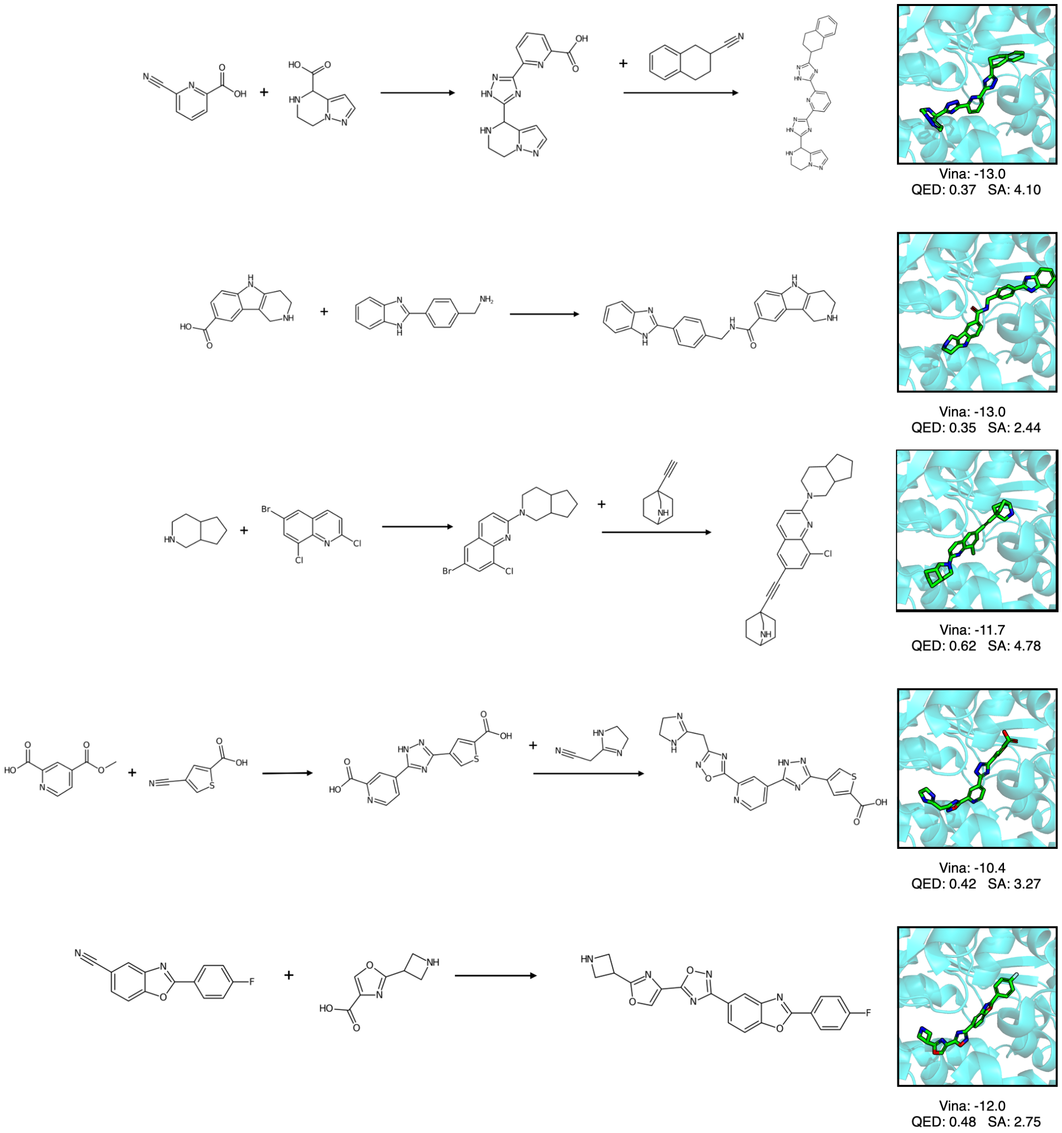

Figure A.7: SynFlowNet-generated molecules and synthesis pathways for SeH.

C.6 Curated building blocks sets given target data

Here we provide further details on the methods and rationale for the fragment-based building block curation strategy outlined in C.4. To enable high molecule purchaseability in a real world drug discovery campaign, we use the Enamine reactions in all these experiments. GPU-accelerated Vina docking was used for reward computation (see App. B.1).

Experimental fragment extration pipeline Fragment-based drug discovery (FBDD) simplifies the search space by focusing on smaller molecular fragments rather than designing or screening full molecules, enabling a more efficient exploration of potential binding interactions (Murray & Rees, 2009; Thomas et al., 2019). To identify molecules bound within the same pocket across different protein structures, we performed a sequence-based search using MMSeqs2 (Steinegger & Söding, 2017) across the Protein Data Bank (PDB). We retained only hits with a sequence identity of 90% or greater and an alignment overlap exceeding 80% to ensure high similarity to the reference target.

The resulting PDB structures were structurally aligned[6] according to the ligand-bound chain with the reference structure. Any ligand with at least one atom within 2 Å of the reference ligand in the binding site was selected for further analysis.

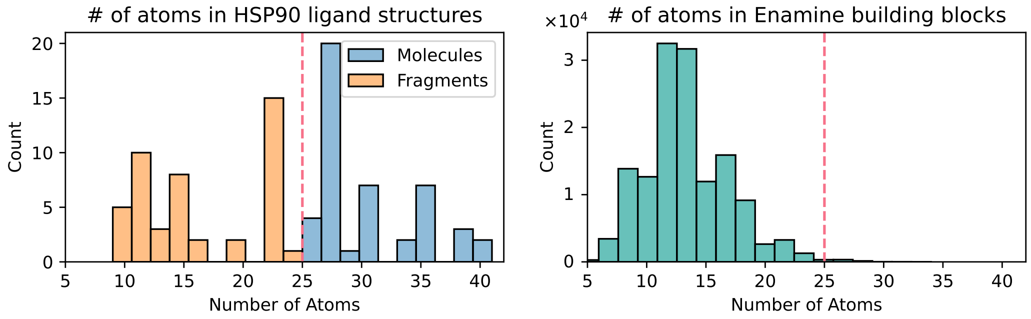

Figure A.8: Curation of HSP90 fragments and building blocks. Red line shows threshold at which we denote a molecule a 'fragment' and was choosen to max the upper limit of the typical BB size.

Selection of small molecule fragments To isolate small molecular fragments suitable for FBDD, we filtered out ligands containing more than 25 atoms, as this atom count aligns with typical building block molecule sizes. This is done for 2 reasons; (i) we can pretend that we are in a drug discovery campaign where a fragment screen has just been conducted and there is no leaked information from 'full' molecules and (ii) this threshold aligns well with the typical Enamine building block in terms of size (Figure A.8).

Curation of Enamine Building Blocks For every experimentally validated fragment, we selected the top 100 closest building block molecules from the Enamine library based on molecular similarity. After removing duplicate SMILES entries, this process resulted in thousands of curated building blocks for each target.

Example fragment screens We study two targets. The first is Heat Shock Protein 90 (HSP90), for which there are a large number of ligand bound structures, mostly thanks to FBDD campaigns conducted in industry (Murray et al., 2010; Woodhead et al., 2010). The second is the SARS-CoV-2 Main Protease (Mpro), for which there is a large amount of structural data and in particular fragments from a fragment screen performed by the COVID Moonshot open-science initiative (Boby et al., 2023). In the case of HSP90, using PDB entry 2XJX (Woodhead et al., 2010) as the reference structure, our ligand extraction pipeline identified 92 molecules. Filtering for small molecular fragments of fewer than 25 atoms further refined this set for analysis. The methodology was repeated for Mpro using PDB entry 7GAW, where we extracted 138 molecules.

Examples of molecules designed for Mpro using our curated building block set are shown in Figure A.9.

Alignment of SynFlowNet designs with experiments Murray et al. (2010) performed a fragment screen of HSP90 and identified two distinct lead classes of binding modes: (i) an aminopyrimidine class that formed hydrogen bonds with an asparagine residue and several conserved water molecules within the pocket and (ii) a phenolic class that primarily binds through water-mediated hydrogen bonds networks. These complicated water-dependent interaction networks are extremely challenging to model computationally and is a clear limitation of Vina docking (which does not take into account waters). Figure A.10 shows that by guiding SynFlowNet in this way, we can design molecules and poses that fit with these experimentally validated interactions.

[6] We use the `superimpose_homologs` function from biotite: `www.biotite-python.org`

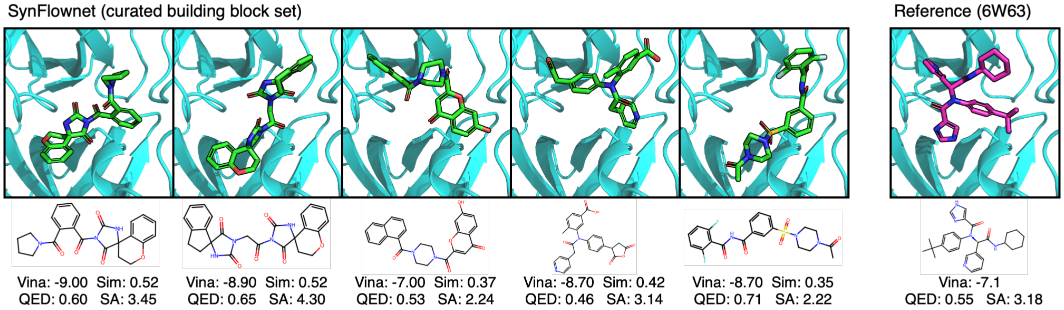

Figure A.9: **Example molecules generated by SynFlowNet for the Mpro case study.** Green molecules are those generated by SynFlowNet. Magenta shows the molecule from the reference structure. 'Sim' is the Tanimoto similarity between a designed molecule and the reference structure.

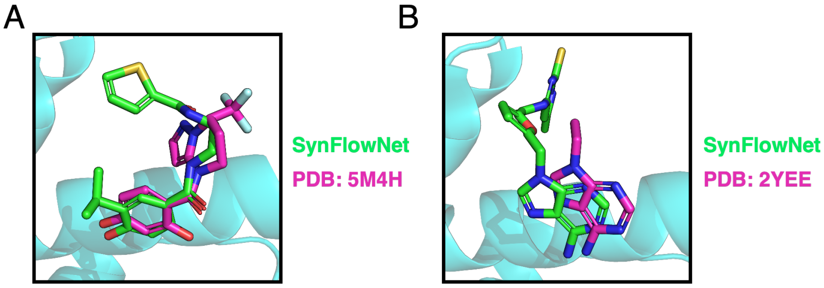

Figure A.10: **SynFlowNet designs molecules that align with real world fragment experiments.** Murray et al. (2010) performed a fragment screen for the HSP90 target and identified two classes of binding modes for the target: those built around (A) phenolic compounds (e.g. PDB:5M4H) and (B) aminopyrimidine compounds (e.g. PDB:2YEE). SynFlowNet trained on a target-curated building block set was able to consistently design compounds based on these experimentally validated binding modes, meaning we can be more certain that they are likely to bind well in reality. Green shows SynFlowNet-designed molecule and magenta are fragments from x-ray crystallography experiments.