DIVERSE HITS IN DE NOVO MOLECULE DESIGN: A DIVERSITY-BASED COMPARISON OF GOAL-DIRECTED GENERATORS

Philipp Renz **Sohvi Luukkonen** **Günter Klambauer**

ABSTRACT

Goal-directed molecular generators have been proposed as a solution to discover novel drug candidates, but often are prone to "mode collapse", which is when they only generate a limited number of similar compounds. The need to generate a diverse set of desired molecules to increase the success chances of drug discovery projects has been identified as a central problem by the research community. However, common benchmarks often lack adequate diversity metrics and overlook the impact of the search budget on model performance. We rectify these two shortcomings, by a) offering a diversity-based evaluation of goal-directed generative models using the principled #Circles metric, and b) evaluating the models under constraints of the number of calls to the scoring functions or the available compute time. Notably, our findings highlight the superior performance of SMILES-based auto-regressive models over graph-based/genetic algorithm counterparts in generating diverse sets of desired compounds.

1 INTRODUCTION

De novo molecule design enables the exploration of the vast chemical space. Goal-directed *de novo* drug design (DNDD) focuses on the generation of molecules possessing specific properties such as efficacy, toxicity, and drug-likeness (Schneider, 2013), by exploring the vast space of drug-like compounds (estimated to contain up to 10^{60} molecules) (Walters, 2019). This is achieved through the generation of novel molecular structures, guided by on-the-fly feedback from a scoring function(s) to incorporate desired properties in a goal-directed generation. Recent years have seen a surge in interest in the field and a range of new methods has been proposed, especially deep learning-based ones (Sanchez-Lengeling & Aspuru-Guzik, 2018; Elton et al., 2019; Luukkonen et al., 2023; Fromer & Coley, 2023).

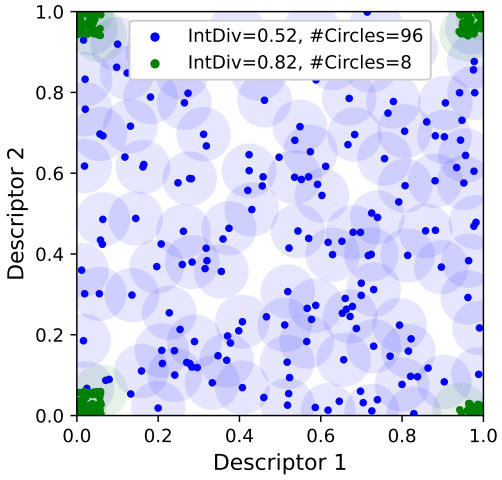

Figure 1: Internal diversity fails to capture the coverage of chemical space and can be optimized by a few clusters of compounds. #Circles accurately captures the coverage of the space based on a relevant distance threshold.

Goal-directed generation and the diversity of generated molecules. Goal-directed DNDD methods usually rely on machine learning-based quantitative structure-property relationship (QSPR) models as scoring functions, which are frequently trained with limited data. As a consequence, these models often yield imperfect approximations and biased outcomes (Renz

et al., 2019). The errors stemming from QSPR models are propagated to molecule generation, introducing a layer of uncertainty that persists throughout the downstream stages of the drug discovery process. Thus, it is crucial to be able to generate a diverse set of high-scoring molecules to increase the chances of finding a successful molecule(s) (Martin, 2001; Seneci, 2002; Angeli & Gaviraghi, 2002; Gorse, 2006; Benhenda, 2017). Furthermore, generating a diverse set of compounds might help create molecules outside the patented chemical space (Shimizu et al., 2023). Unfortunately, many proposed generative models are prone to "mode collapse", where they only generate a small number of similar compounds. This issue has been addressed with various approaches aiming to improve the diversity of the generated molecules (Rupakheti et al., 2015; Liu et al., 2019; Chen et al., 2020; Blaschke et al., 2020; Bengio et al., 2021; Pereira et al., 2021; Bjerrum et al., 2023).

Previous comparison studies used insufficient diversity metrics. Well-known DNDD benchmarking platforms and leaderboards, such as GuacaMol (Brown et al., 2019) and MOSES (Polykovskiy et al., 2020), include some classic diversity metrics: uniqueness and/or internal diversity, in the case of non-goal-directed compound generation. But to our knowledge, there has not been a systematic benchmark study of the capacity of different goal-based generative models to generate a diverse set of high-scoring molecules. Moreover, these traditional metrics exhibit significant limitations in characterizing well the chemical space represented by a set of compounds (Waldman et al., 2000; Xie et al., 2023). For example, Figure 1 shows how the arguably most commonly used diversity metric, *internal diversity*, fails to capture coverage of chemical space. Even more rudimental metrics, such as the fraction of unique compounds and unique Bemis-Murcko scaffolds (Bemis & Murcko, 1996) are also not well-suited as they can be optimized by generating many highly similar compounds, differing only in single atoms or bonds. Recently several diversity metrics based on sphere exclusion algorithm (Willett, 2001) have been proposed to quantify the chemical space coverage: *SEDiv* by Thomas et al. (2021) and *#Circles* by Xie et al. (2023). These metrics were shown to align well with chemical intuition regarding the chemical diversity of known libraries, and shown to be highly correlated to coverage of biological functionalities.

Generative models should be evaluated and compared under a fixed computational budget. Practically relevant measures are the overall compute time or the number of scoring function evaluations. The latter is of special importance as increasing the number of evaluation calls can lead to overfitting to biased QSPR models and decrease the quality of the generated molecules over time. Moreover, given reported failure modes (Renz et al., 2019) when using machine learning-based scoring functions the field has shown interest in replacing them with more costly physics-based methods, such as docking (Thomas et al., 2021; Guo et al., 2021; Goel et al., 2021; Elend et al., 2022). Sample efficient methods are important for successful applications of such expensive scoring functions. On the other end of the spectrum, compute-efficient generators are needed for replacing large-scale virtual screening campaigns using less costly scoring functions. Gao et al. (2022) tested the sample efficiency of a range of generative models given a constraint on the number of scoring function calls, but no studies exist that focus the diversity of the generated compounds under such constraints.

Contributions. In this work, we address the two shortcomings of previous comparisons, a) the insufficient diversity metrics and b) the generation without limitations on the computational budget. We systematically benchmark the performance of established generative algorithms at generating diverse high-scoring molecules, referred to as *diverse hits*. The evaluation is conducted within the framework of goal-directed optimization, in which the algorithms operate under the constraint of a limited number of scoring function calls or one of limited time, giving more emphasis to the computational cost of the generator. We utilize the *#Circles* diversity metric as a key performance indicator, providing a comprehensive assessment of the efficiency of generative models in real-world scenarios.

2 Benchmark Setup

2.1 Molecular Diversity

In this study, we adopt the diversity metric, #Circles, proposed by Xie et al. (2023), which aims to maximize the number of high-scoring compounds that are sufficiently distinct to potentially exhibit different bioactivity profiles. Given a scoring function $s(m)$ that assigns a score to a molecule m, the

generative models are tasked with generating a set of high-scoring molecules $\mathcal{G}$, where all molecules surpass a score threshold S, $s(m) \geq S$ for all $m \in \mathcal{G}$. The algorithm's final performance is evaluated based on the diversity of these high-scoring compounds according to the #Circles metric,

$$\mu(\mathcal{G}; D) = \max_{\mathcal{C} \in \mathcal{P}(\mathcal{G})} |\mathcal{C}| \text{ s.t. } \forall x \neq y \in \mathcal{C} : d(x, y) \geq D,$$

where $\mathcal{P}$ denotes the power set, $d(x, y)$ represents the distance between compounds x and y, and D is a distance threshold. This metric, referred to as the number of *diverse hits*, signifies the size of the largest subset of $\mathcal{G}$ such that all compounds have pairwise distances greater than D.

In this study, we use the Tanimoto distance between Morgan fingerprints (radius=2, size=2048), and a distance threshold of $D = 0.7$, as it aligns well with the sharp drop in the probability of similar bioactivities beyond this value (Jasial et al., 2016; Sayle, 2019; Landrum). Although exact computation of this metric is NP-complete, we can efficiently approximate it using the MaxMin algorithm (Sayle, 2019).

2.2 Scoring functions

Bioactivity prediction models. We evaluate the methods on three well-established molecule bioactivity optimization tasks: the JNK3 and GSK3β tasks studied in (Li et al., 2018), and the DRD2 task introduced in (Blaschke et al., 2020). Each task is based on a data set of compounds and associated binary bioactivity labels for the respective targets. We partition the data into training and testing sets using a random 75/25% split. For each task, we train a Random Forest classifier (Breiman, 2001) on Morgan fingerprints (radius=2, size=2048) (Rogers & Hahn, 2010). Table A1 provides details about the datasets and the performance of the predictive models. All scoring functions exhibit robust predictive performance, as indicated by their ROCAUC and Average Precision (AP) values.

During optimization, we use the RF classifier's probabilistic output for a compound being active, $p_{\text{RF}}(s)$, as a scoring function. Whereas when predicting if a compound is a hit, we adopt a score threshold of $S = 0.5$ as this results in high precision values, while still maintaining acceptable recall. A higher score threshold would result in higher precision but might result in biasing the generator towards recovering active compounds from the training set (Renz et al., 2019), and discarding many potentially active compounds. Further details on the QSAR models are given in Section A.1.

Property filters. Generative models often generate compounds with very high molecular weights (MW) or water-octanol partition coefficients (logP) and may contain idiosyncratic substructures, rendering them impractical for drug discovery projects (Renz et al., 2019; Thomas et al., 2022b). This poses a challenge to model evaluations, as these compounds would likely be discarded in real-world applications. To address this issue, inspired by (Thomas et al., 2022b), we incorporate lenient property constraints into the scoring functions. We determined quantile-based lower and upper bounds for both MW and logP, ensuring that 99% of the compounds within the GuacaMol dataset (Brown et al., 2019) fall within these limits. Additionally, to avoid compounds with idiosyncratic substructures, we calculate the fraction of unobserved fingerprint bits unobserved in the reference set for each compound in a calibration set. We determine an upper bound for this fraction such that 99% of the calibration set falls below the bound. Further details on these filters are given in Section A.2. The scores of compounds violating any of these filters are set to zero.

Diversity filter. We equip all tested models with the *diversity filter* (DF) proposed by Blaschke et al. (2020) to enable generation of diverse compounds. This filter assigns zero scores to compounds that are within a distance threshold $D_{\text{DF}} = 0.7$ to previously found compounds surpassing the score threshold S. This ensures that the optimization process does not get stuck in local optima, but instead explores new regions of the chemical space. Preliminary experiments have shown that the DF improves performance of generative algorithms originally designed for single molecule optimization and allows for a meaningful inclusion of those algorithms in our benchmark. A more detailed description of the DF is given in Section A.3

The final scoring function is the product of the bioactivity prediction model, the property filters, and the diversity filter.

2.3 COMPUTE CONSTRAINTS

We evaluate the performance of the generators to create diverse hits under two computational constraint settings: (a) **Sample limit**, we limit the number of scoring function evaluations to 10K as proposed in (Gao et al., 2022), and (b) **Time limit**, we limit the time available to the algorithms to 600 seconds. All algorithms are executed using 8 cores of an AMD Ryzen Threadripper 1920X and a single NVidia RTX 2080 GPU.

2.4 GENERATIVE MODELS

We utilize our benchmark setup to assess the following 12 methods. The methods were chosen based on their performance in previous benchmarks (Brown et al., 2019; Gao et al., 2022) and to ensure that a range of methodically different approaches is included.

We test six LSTM-based auto-regressive models operating on SMILES: **LSTM-HC** (Segler et al., 2018) optimizing with a hill-climb algorithm, **LSTM-PPO** (Neil et al., 2018) optimizing with the PPO algorithm, **Reinvent** (Olivecrona et al., 2017) optimizing with a modified REINFORCE algorithm, and three extensions of Reinvent: **AugmentedHC** (mixture of Reinvent and hill-climb) (Thomas et al., 2022a), **AugMemory** (Guo & Schwaller, 2023), and BestAgentReminder (**BAR**) method (Atance et al., 2022). We also test three genetic algorithms making use of mutations of different molecular representations: **GraphGA** (Jensen, 2019) operating molecular graphs, **SmilesGA** (Yoshikawa et al., 2018) operating on SMILES, and **Stoned** (Nigam et al., 2021) operating on SELFIES. We further test three models that generate molecules via sequential graph edits: **Mars** (Xie et al., 2021), **Mimosa** (Fu et al., 2022), and **GFlowNet/GFlowNetDF** (Bengio et al., 2021), which is tested with and without the DF as it supports diverse generation by default. Further details about these methods and the choice to exclude others are discussed in B.

We compare the tested methods against two virtual screening baselines using the GuacaMol dataset (Brown et al., 2019) as a library. The first variant, **VS Random**, scans the library in random order, scoring each compound using the respective scoring function. For the second variant **VS MaxMin** we first use the MaxMin algorithm (Sayle, 2019) to sort the library. By doing so we ensure that the algorithm first screens the most diverse compounds in the library.

2.5 OPTIMIZATION

We carried out a hyperparameter search to find the best settings for each combination of generative algorithm, scoring function, and computational constraint. Employing a random search with 15 trials for each combination, we explored various hyperparameter ranges, and the selected hyperparameters are detailed in Table C1. We executed five independent runs, each initialized with a distinct random seed.

Throughout the optimization process, we tracked all generated compounds, their corresponding scores, and the time of generation. The recording of all scored compounds is essential when considering search efficiency, ensuring that no potentially valuable compounds are needlessly discarded. This is especially important when using a diversity filter, as the search steers away from already discovered solutions, as they are not accessible by sampling the final model.

3 RESULTS AND DISCUSSION

In this work, we benchmarked the capacity of a wide range of goal-directed molecular generators to design diverse hits under two computational constraint settings for three protein targets. The main results in terms of the number of diverse hits under the sample and time constraints are shown in Figure 2 and discussed here. Extended results with additional metrics are given in Section D.

Large differences in performance between models and optimization tasks. Above all, we observe in Figure 2 a significant difference in the capacity to produce diverse hits between the different algorithms: the number of diverse hits ranges from several compounds for Mars (worst) to several hundred compounds for AugMemory (best) in the sample constrained setting. We also see that the

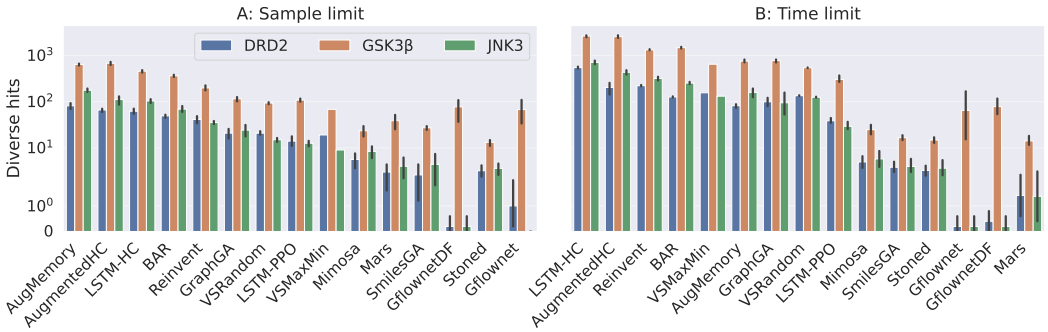

Figure 2: Number of diverse hits found for each optimization task under the given compute constraints. The results span multiple orders of magnitude and the order of the generator (avg. rank) depends on the constraint type. SMILES-based LSTM models perform best in generating diverse hits. Error bars show the range of the results.

performance is highly task-dependent: most approaches find $\sim 10\times$ more diverse hits for GSK3β than DRD2, with the biggest differences showing for the GFlowNet generators.

SMILES-based LSTM models perform best in generating diverse hits. Generally, the top ranks are dominated by auto-regressive SMILES-based models. In the sample limit setting, AugMemory performs the best. It uses experience replay with selective purge and data augmentation allowing it to outperform its parent model Reinvent. Similarly, the AugmentedHC model can also outperform its parent models Reinvent and LSTM-HC as shown in the original paper (Thomas et al., 2022a). LSTM-HC attains the third rank and can outperform Reinvent, which is in contrast to results in single compound optimization tasks (Gao et al., 2022). Also, the BAR model slightly outperforms its base model Reinvent. Whereas LSTM-PPO performs relatively poorly compared to the other SMILES-based models. The increase in performance of the extensions compared to parent methods, comes with a significant computational cost as under the time limit both LSTM-HC and Reinvent outperform their extensions.

Limited number of diverse hits with graph-based and genetic algorithms. The graph-based algorithms generally occupy the lower ranks in this comparison. GraphGA is the only graph-based model that outperforms the virtual screening baselines. SmilesGA and Stoned both perform poorly in this comparison, which is in contrast to the single compound optimization task under the same sample constraint studied in (Gao et al., 2022). We also found Mars and GFlowNet to perform poorly in this comparison, although performing well in previous studies (Bengio et al., 2021; Xie et al., 2023; 2021). We think that this is because in these studies these models were trained using up to 1M scoring function evaluations. Our results, however, are in line with the findings of Gao et al. (2022) who found that these models perform poorly when the sample budget is limited.

4 Conclusion

In this work, we presented a diversity-based comparison of goal-directed molecule generators under computational constraints that amends two shortcomings of previous benchmarks: the use of insufficient diversity metrics and the generation without limitations on the computational budget. Under computational constraints, either in compute time or in the number scoring function calls, SMILES-based auto-regressive models are able to generate a large number of diverse hits, while graph-based models exhibit limited success.

We found that model performance often does not translate between different optimization dimensions like single/diverse molecule generation, or time/sample constraints. The large differences between the ability of different models to generate diverse hits emphasize the importance of careful model selection based on relevant computational constraints and diversity metrics.

REFERENCES

P. Angeli and G. Gaviraghi. Chemical and biological diversity in drug discovery. In *Pharmacochemistry Library*, volume 32, pp. 95–96. 2002. ISBN 978-0-444-50760-0. doi: 10.1016/S0165-7208(02)80011-3.

Sara Romeo Atance, Juan Viguera Diez, Ola Engkvist, Simon Olsson, and Rocío Mercado. De Novo Drug Design Using Reinforcement Learning with Graph-Based Deep Generative Models. *J. Chem. Inf. Model.*, 62(20):4863–4872, October 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.2c00838.

Guy W. Bemis and Mark A. Murcko. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.*, 39(15):2887–2893, January 1996. ISSN 0022-2623. doi: 10.1021/jm9602928.

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

Mostapha Benhenda. ChemGAN challenge for drug discovery: Can AI reproduce natural chemical diversity? *arXiv:1708.08227 [cs stat]*, August 2017.

Esben Jannik Bjerrum, Christian Margreitter, Thomas Blaschke, and Raquel López-Ríos de Castro. Faster and more diverse de novo molecular optimization with double-loop reinforcement learning using augmented SMILES. *J Comput Aided Mol Des*, 37(8):373–394, August 2023. ISSN 0920-654X, 1573-4951. doi: 10.1007/s10822-023-00512-6.

Thomas Blaschke, Ola Engkvist, Jürgen Bajorath, and Hongming Chen. Memory-assisted reinforcement learning for diverse molecular de novo design. *J Cheminform*, 12(1):68, December 2020. ISSN 1758-2946. doi: 10.1186/s13321-020-00473-0.

Leo Breiman. Random Forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324.

Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.*, 59(3):1096–1108, March 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.8b00839.

Binghong Chen, Tianzhe Wang, Chengtao Li, Hanjun Dai, and Le Song. Molecule Optimization by Explainable Evolution. In *ICLR*, October 2020.

Lars Elend, Luise Jacobsen, Tim Cofala, Jonas Prellberg, Thomas Teusch, Oliver Kramer, and Ilia A. Solov'yov. Design of SARS-CoV-2 Main Protease Inhibitors Using Artificial Intelligence and Molecular Dynamic Simulations. *Molecules*, 27(13):4020, January 2022. ISSN 1420-3049. doi: 10.3390/molecules27134020. URL https://www.mdpi.com/1420-3049/27/13/4020.

Daniel C. Elton, Zois Boukouvalas, Mark D. Fuge, and Peter W. Chung. Deep learning for molecular design—a review of the state of the art. *Mol. Syst. Des. Eng.*, 4(4):828–849, August 2019. ISSN 2058-9689. doi: 10.1039/C9ME00039A.

Jenna C. Fromer and Connor W. Coley. Computer-aided multi-objective optimization in small molecule discovery. *Patterns*, 4(2):100678, February 2023. ISSN 2666-3899. doi: 10.1016/j.patter.2023.100678. URL https://www.sciencedirect.com/science/article/pii/S2666389923000016.

Tianfan Fu, Cao Xiao, Xinhao Li, Lucas M. Glass, and Jimeng Sun. MIMOSA: Multi-constraint Molecule Sampling for Molecule Optimization, February 2022.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization, June 2022.

Manan Goel, Shampa Raghunathan, Siddhartha Laghuvarapu, and U. Deva Priyakumar. MoleGuLAR: Molecule Generation Using Reinforcement Learning with Alternating Rewards. *J. Chem. Inf. Model.*, 61(12):5815–5826, December 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c01341. URL https://doi.org/10.1021/acs.jcim.1c01341.

Alain-Dominique Gorse. Diversity in Medicinal Chemistry Space. *Curr. Top. Med. Chem.*, 6(1): 3–18, January 2006.

David E. Graff, Eugene I. Shakhnovich, and Connor W. Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chem. Sci.*, 12(22):7866–7881, 2021. ISSN 2041-6520, 2041-6539. doi: 10.1039/D0SC06805E.

Jeff Guo and Philippe Schwaller. Augmented Memory: Capitalizing on Experience Replay to Accelerate De Novo Molecular Design, May 2023.

Jeff Guo, Jon Paul Janet, Matthias R. Bauer, Eva Nittinger, Kathryn A. Giblin, Kostas Papadopoulos, Alexey Voronov, Atanas Patronov, Ola Engkvist, and Christian Margreitter. DockStream: a docking wrapper to enhance de novo molecular design. *J. Cheminformatics*, 13(1):89, November 2021. ISSN 1758-2946. doi: 10.1186/s13321-021-00563-7. URL https://doi.org/10.1186/s13321-021-00563-7.

Swarit Jasial, Ye Hu, Martin Vogt, and Jürgen Bajorath. Activity-relevant similarity values for fingerprints and implications for similarity searching. *F1000Res*, 5:Chem Inf Sci–591, April 2016. ISSN 2046-1402. doi: 10.12688/f1000research.8357.2.

Jan H. Jensen. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.*, 10(12):3567–3572, March 2019. ISSN 2041-6539. doi: 10.1039/C8SC05372C.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-Objective Molecule Generation using Interpretable Substructures, July 2020.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Self-Referencing Embedded Strings (SELFIES): A 100
arXiv:1905.13741 [phys. phys.:quant-ph stat], March 2020.

Greg Landrum. A new "Lessel and Briem like" dataset.

Greg Landrum. RDKit: Open-source cheminformatics. http://www.rdkit.org, 2006.

Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *J. Cheminformatics*, 10(1):33, July 2018. ISSN 1758-2946. doi: 10.1186/s13321-018-0287-6.

Xuhan Liu, Kai Ye, Herman W. T. van Vlijmen, Adriaan P. IJzerman, and Gerard J. P. van Westen. An exploration strategy improves the diversity of de novo ligands using deep reinforcement learning: A case for the adenosine A2A receptor. *J. Cheminformatics*, 11(1):35, May 2019. ISSN 1758-2946. doi: 10.1186/s13321-019-0355-6.

Sohvi Luukkonen, Helle W. van den Maagdenberg, Michael T. M. Emmerich, and Gerard J. P. van Westen. Artificial intelligence in multi-objective drug design. *Curr. Opin. Struct. Biol.*, 79:102537, April 2023. ISSN 0959-440X. doi: 10.1016/j.sbi.2023.102537. URL https://www.sciencedirect.com/science/article/pii/S0959440X23000118.

Yvonne C. Martin. Diverse Viewpoints on Computational Aspects of Molecular Diversity. *J. Comb. Chem.*, 3(3):231–250, May 2001. ISSN 1520-4766. doi: 10.1021/cc000073e.

Daniel Neil, Marwin Segler, Laura Guasch, Mohamed Ahmed, Dean Plumbley, Matthew Sellwood, and Nathan Brown. Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design. https://openreview.net/forum?id=HkcTe-bR-, February 2018.

AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alán Aspuru-Guzik. Beyond generative models: Superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chem. Sci.*, 12(20):7079–7090, May 2021. ISSN 2041-6539. doi: 10.1039/D1SC00231G.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *J. Cheminformatics*, 9(1):48, September 2017. ISSN 1758-2946. doi: 10.1186/s13321-017-0235-x.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(85):2825–2830, 2011.

Tiago Pereira, Maryam Abbasi, Bernardete Ribeiro, and Joel P. Arrais. Diversity oriented Deep Reinforcement Learning for targeted molecule generation. *J Cheminform*, 13(1):21, December 2021. ISSN 1758-2946. doi: 10.1186/s13321-021-00498-z.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front. Pharmacol.*, 11, 2020. ISSN 1663-9812.

Philipp Renz, Dries Van Rompaey, Jörg Kurt Wegner, Sepp Hochreiter, and Günter Klambauer. On failure modes in molecule generation and optimization. *Drug Discov. Today: Technol.*, 32–33: 55–63, December 2019. ISSN 1740-6749. doi: 10.1016/j.ddtec.2020.09.003.

David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.*, 50(5): 742–754, May 2010. ISSN 1549-9596. doi: 10.1021/ci100050t.

Chetan Rupakheti, Aaron Virshup, Weitao Yang, and David N. Beratan. Strategy To Discover Diverse Optimal Molecules in the Small Molecule Universe. *J. Chem. Inf. Model.*, 55(3):529–537, March 2015. ISSN 1549-9596. doi: 10.1021/ci500749q.

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aat2663.

Roger Sayle. 2D Similarity, Diversity and Clustering in RDKit, 2019. URL https://www.nextmovesoftware.com/talks/Sayle_2DSimilarityDiversityAndClusteringInRdkit_RDKITUGM_201909.pdf.

Gisbert Schneider. *De Novo Molecular Design*. 2013. ISBN 978-3-527-67701-6. doi: 10.1002/9783527677016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.

Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.*, 4(1): 120–131, January 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00512.

Pierfausto Seneci. Chemical diversity as a driving force to design and put in practice synthetic strategies leading to combinatorial libraries for lead discovery and lead optimization. In Henk van der Goot (ed.), *Pharmacochemistry Library*, volume 32 of *Trends in Drug Research III*, pp. 147–160. January 2002. doi: 10.1016/S0165-7208(02)80016-2.

Yugo Shimizu, Masateru Ohta, Shoichi Ishida, Kei Terayama, Masanori Osawa, Teruki Honma, and Kazuyoshi Ikeda. AI-driven molecular generation of not-patented pharmaceutical compounds using world open patent data. *J. Cheminformatics*, 15(1):120, December 2023. ISSN 1758-2946. doi: 10.1186/s13321-023-00791-z. URL https://doi.org/10.1186/s13321-023-00791-z.

Morgan Thomas, Robert T. Smith, Noel M. O'Boyle, Chris de Graaf, and Andreas Bender. Comparison of structure- and ligand-based scoring functions for deep generative models: A GPCR case study. *J. Cheminformatics*, 13(1):39, May 2021. ISSN 1758-2946. doi: 10.1186/s13321-021-00516-0.

Morgan Thomas, Noel M. O'Boyle, Andreas Bender, and Chris de Graaf. Augmented Hill-Climb increases reinforcement learning efficiency for language-based de novo molecule generation. *J. Cheminformatics*, 14(1):68, October 2022a. ISSN 1758-2946. doi: 10.1186/s13321-022-00646-z.

Morgan Thomas, Noel M. O'Boyle, Andreas Bender, and Chris De Graaf. Re-evaluating sample efficiency in de novo molecule generation, December 2022b.

Austin Tripp, Gregor N C Simm, and José Miguel Hernández-Lobato. A Fresh Look at De Novo Molecular Design Benchmarks.

Marvin Waldman, Hong Li, and Moises Hassan. Novel algorithms for the optimization of molecular diversity of combinatorial libraries11Color Plates for this article are on pages 533–536. *J. Mol. Graph. Model.*, 18(4):412–426, January 2000. ISSN 1093-3263. doi: 10.1016/S1093-3263(00)00071-1.

W. Patrick Walters. Virtual Chemical Libraries. *J. Med. Chem.*, 62(3):1116–1124, February 2019. ISSN 0022-2623. doi: 10.1021/acs.jmedchem.8b01048.

David Weininger. SMILES, a chemical language and information system. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, February 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005.

Valerie J. Gillet Willett, Peter. Dissimilarity-Based Compound Selection for Library Design. In *Combinatorial Library Design and Evaluation*. 2001. ISBN 978-0-429-18097-2.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn*, 8(3-4):229–256, May 1992. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00992696.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. MARS: Markov Molecular Sampling for Multi-objective Drug Discovery. *arXiv:2103.10432 [cs q-bio]*, March 2021.

Yutong Xie, Ziqiao Xu, Jiaqi Ma, and Qiaozhu Mei. How Much Space Has Been Explored? Measuring the Chemical Space Covered by Databases and Machine-Generated Molecules. In *ICLR*, February 2023.

Naruki Yoshikawa, Kei Terayama, Teruki Honma, Kenta Oono, and Koji Tsuda. Population-based de novo molecule generation, using grammatical evolution. *arXiv:1804.02134 [phys. q-bio]*, April 2018.

A SCORING FUNCTION DETAILS

A.1 DATASETS AND PREDICTIVE MODELS

We test the generative models using three established optimization tasks: the JNK3 and GSK3β tasks used in (Li et al., 2018; Jin et al., 2020; Xie et al., 2023), and the DRD2 task introduced in (Blaschke et al., 2020). All three tasks are based on datasets of compounds with associated binary bioactivity labels. The label indicates whether a compound is active or inactive against the respective target. The data is partitioned into training and testing sets using a 75/25% split. For each dataset, we train a Random Forest classifier (Breiman, 2001) using the implementation provided by scikit-learn (Pedregosa et al., 2011). The classifier is trained on ECFP4 fingerprints (Rogers & Hahn, 2010) of size 2048 as implemented in (Landrum, 2006). The predicted probability of a compound being active serves as the foundation for the scoring function and will be augmented by the property and diversity filters.

Table A1: Performance Metrics for JNK3, GSK3β and DRD2 activity prediction models. The table shows the ROCAUC, Average Precision (AP), Precision, and Recall at a 0.5 threshold, average scores of train and test actives, the number of samples, and the number of diverse train actives at a 0.7 distance threshold.

Target	ROCAUC	AP	Prec@0.5	Rec@0.5	Prec@0.9	Rec@0.9	#Samples	#Actives@0.70
JNK3	0.96	0.86	0.97	0.62	0.98	0.21	50390	220
GSK3	0.98	0.93	0.98	0.72	0.99	0.32	52802	643
DRD2	1.00	0.91	0.89	0.79	0.97	0.30	102981	229

Table A1 shows the classification performance of the predictive models on the respective test sets. We evaluated the classification performance for the classifiers using the Area under the Receiver Operating Characteristic Curve (ROCAUC) and Average Precision (AP) metrics, which show good performance for all three datasets. We also report the Precision and Recall at the score threshold, $S = 0.5$, used in the molecule optimization tasks. Table A1 also shows the number of diverse training actives at a distance threshold, $D = 0.7$, giving an indication of at least how many diverse actives the generative models should be able to find.

We establish a score threshold of $S = 0.5$ for the optimization tasks. Precision values are high at this threshold, indicating that compounds scoring above this mark are very likely to be true actives. At the same time, the recall values are not excessively low, indicating that not too many true actives are rejected. Increasing the score threshold to a higher value like 0.9 would result in marginally higher precision values but drastically reduced recall, and would result in discarding many potentially active compounds. Furthermore using a high score threshold biases the optimization process towards recovering active compounds in the training set (Renz et al., 2019).

A.2 PROPERTY FILTERS

Generative models have been observed to frequently produce compounds that feature atypical substructures or yield compounds with exceptionally high molecular weight (MW) or water-octanol partition coefficients (logP) values (Renz et al., 2019; Thomas et al., 2022b). To enable a meaningful comparison we constrain the optimization objective to ensure that the generated molecules have properties within the range of those in a reference set. We use the ChEMBL subset provided in GuacaMol(Brown et al., 2019) as a reference set. In line with (Thomas et al., 2022b), we use relatively lenient property constraints to ensure that the optimization objective is not overly restrictive but ensures that compounds with strongly atypical properties are not rewarded. For the molecular weight (MW) and water-octanol partition coefficient (logP), we determine two-sided quantile-based lower and upper bounds, ensuring that 99% of the compounds in the ChEMBL dataset fall within these limits. For MW this results in a permissible range of $[157, 761]$ Da, and for logP this range is $[-2.0, 8.3]$.

We adopt the methodology outlined by Thomas et al. (2022b) to address the challenge of uncommon substructures. We first partition the reference set into a training set and a calibration set, comprising 1.3M and 300K compounds, respectively. Then, we compute the unfolded ECFP4 fingerprints

(Rogers & Hahn, 2010) for all compounds within the training set, and determine the set of all occurring hash values. This gives a reference of substructures typically found in drug-like molecules. We then compute the fingerprints for each compound in the calibration set and calculate the proportion of substructures absent in the training set. We determine a threshold for this proportion such that 99% of the compounds in the calibration set fall below it, which evaluates to 0.08.

A.3 DIVERSITY FILTER

Most generative models have been designed to find single top-scoring compounds and are not incentivized to find diverse solutions. Given the complexity of the optimization objective, it is hard to directly optimize it. The diversity filter (DF) proposed in (Blaschke et al., 2020) closely reflects the optimization objective and can be easily incorporated into generative models, by incorporating it into the used scoring function.

The DF is initialized using an empty list M. Whenever a compound c is generated, it is compared with all compounds in M. If its distance to any compound in M is less than D_{DF} the compound does not pass the diversity filter and its DF-score is set to zero. If a compound passes the diversity filter, its DF-score is set to one. If a compound passes the diversity filter and $s(c) > s_{\text{DF}}$ it is added to M. This procedure is summarized in Algorithm 1.

For the experiments in this study, we set $D_{\text{DF}} = 0.7$ and $s_{\text{DF}} = 0.5$. We do not make explicit use of the bucket mechanism proposed in (Blaschke et al., 2020). Instead, we set the bucket size to one, as this resulted in quicker reorientation and faster exploration in preliminary experiments.

Algorithm 1: Computation of DF-score

Input : Generated compound c, score of compound $s(c)$, Filter list M, DF score threshold s_{DF}, DF distance threshold D_{DF}
Output: DF score s_{DF}, Updated filter list M

```
passes ← True ;                          // Initialize pass status to True
for each compound m in M do
    if distance(c, m) < D_DF then
        passes ← False ;                 // Set pass status to False
        break
if passes and s(c) > S then
    Add c to M ;                         // Add compound to filter list
return passes, M
```

B GENERATIVE MODEL DETAILS

The tested models include a range of auto-regressive models operating on SMILES strings (Weininger, 1988). **LSTM-HC** (Segler et al., 2018) uses a hill-climb algorithm for fine-tuning a pre-trained LSTM model. **Reinvent** (Olivecrona et al., 2017) optimizes a recurrent neural network using the REINFORCE algorithm (Williams, 1992) combined with prior regularization. **AugmentedHC** (Thomas et al., 2022a) forms a hybrid between the Reinvent and LSTM-HC models, by only using the Reinvent loss of the k top-scoring compounds to update the model. **AugMemory** (Guo & Schwaller, 2023) extends Reinvent, adding experience replay and data augmentation to increase sample efficiency. It makes use of selective memory purge to make experience replay compatible with the diversity filter described below. The BestAgentReminder (**BAR**) method (Atance et al., 2022) also extends the Reinvent algorithm, by keeping track of the best agent found so far and intersperses samples from the best agent with samples from the current model to stabilize training. **LSTM-PPO** (Neil et al., 2018) makes use of the popular PPO reinforcement learning algorithm to tune the model (Schulman et al., 2017).

We test three genetic algorithms making use of mutations of different molecular representations. **GraphGA** (Jensen, 2019) is a graph-based genetic algorithm that operates on the graph representation of molecules, and has shown competitive performance in previous benchmarks (Brown et al., 2019; Gao et al., 2022). **SmilesGA** (Yoshikawa et al., 2018) generates novel molecules by encoding

SMILES strings (Weininger, 1988) into production rules of a context-free grammar and randomly inserting mutations into these rules. **Stoned** (Nigam et al., 2021) generates molecules by introducing point mutations into the SELFIES (Krenn et al., 2020) representation of molecules.

We further test a range of models that generate molecules via sequential graph edits. **Mars** (Xie et al., 2021) makes use of Markov chain Monte Carlo sampling to generate molecules and achieved the best performance in a previous diverse optimization evaluation (Xie et al., 2023). **Mimosa** (Fu et al., 2022) also operates on the graph representation of molecules and evolves molecules by applying a sequence of graph edits. **GFlowNet** (Bengio et al., 2021) similarly sequentially builds molecules by graph edits but uses a specialized learning objective to enable diverse candidate generation.

While a range of strategies to promote diversity has been suggested (Pereira et al., 2021; Bjerrum et al., 2023; Blaschke et al., 2020; Liu et al., 2019; Xie et al., 2023; Chen et al., 2020), the diversity filter proposed in (Blaschke et al., 2020) is emerging as a standard approach employed in many studies (Guo & Schwaller, 2023; Thomas et al., 2022b;a; Bjerrum et al., 2023) as it is a natural solution in line with our optimization objective, and is easy to combine with different generative models. Therefore we focus on the use of the DF for inducing diversity into the generated molecules. As GFlowNet is designed for diverse optimization we test it both with and without the diversity filter.

We do not test some other methods designed for diverse optimization, as they are conceptually similar to prior regularization used in Reinvent and its derivatives. We provide a detailed discussion in Section B.1.

B.1 Choice of tested algorithms

We did not test the following algorithms that have been reported to help improve the diversity of generated molecules. We did not include double-loop reinforcement learning (Bjerrum et al., 2023) as it is conceptually very similar to the Augmented Memory algorithm (Guo & Schwaller, 2023). Both algorithms use augmented versions of previously generated compounds to update the generative model multiple times.

We did not test the exploration approaches taken in (Pereira et al., 2021; Liu et al., 2019) as they are conceptually similar to the prior regularization approaches that are used in Reinvent (Olivecrona et al., 2017) and its descendants Augmented Hill-Climb (Thomas et al., 2022a) and Augmented Memory (Guo & Schwaller, 2023). We also did not include the the method proposed in (Rupakheti et al., 2015) in our experiments, as it is similar to the genetic algorithm equipped with the diversity filter.

Active learning methods have been shown to be effective in increasing the sample efficiency in optimization tasks (Graff et al., 2021; Tripp et al.). Testing these methods is beyond the scope of this study, as implementation and tuning is non-trivial. However, learning a fast proxy scoring function effectively reduces the cost of scoring molecules. Therefore the time constraint experiments give an indication of the performance of active learning methods. In principle, all the methods tested in this study can be combined with active learning methods, and pose an interesting avenue for future research.

C Hyperparameter optimization

For each generative model, we performed hyperparameter optimization to identify the best performing hyperparameters for each combination of a generative algorithm, scoring function and the used compute constraint. For each combination we performed 15 runs with independently sampled hyperparameters. The hyperparameter distributions used for the random search and the selected parameters are given in Table C1.

In principle, the performance of the Reinvent derivatives AugmentedHC, AugmentedMem, and BAR could match that of Reinvent, when selecting the right hyperparameters. However, in this comparison, we restricted the parameter ranges to ensure non-trivial differences between the algorithms. This allows us to analyze the impact of the modifications in this setting.

Table C1: Hyperparameter ranges for the tested optimizers. We executed a random search using the distributions specified in this table. The selected hyperparameters for the respective constraint settings and targets are shown in the last six columns.

		Limit	Samples			Time		
Optimizer	Parameter	Search Space	DRD2	GSK3β	JNK3	DRD2	GSK3β	JNK3
AugHC	batch_size	RandInt(128, 512)	482	305	510	440	487	321
	learning_rate	LogUniform($10^{-4}, 10^{-3}$)	3.55e−4	2.89e−4	3.39e−4	2.24e−4	2.65e−4	1.90e−4
	sigma	Uniform(100.0, 500.0)	432.90	412.21	468.84	201.75	358.36	183.90
	topk	Uniform(0.15, 0.35)	0.16	0.17	0.17	0.20	0.24	0.17
AugMemory	augmentation_rounds	RandInt(1, 7)	6	6	2	4	1	1
	batch_size	RandInt(32, 128)	110	126	41	114	125	91
	learning_rate	LogUniform($10^{-4}, 10^{-3}$)	1.56e−4	1.25e−4	2.00e−4	1.61e−4	7.55e−4	3.90e−4
	replay_buffer_size	RandInt(32, 128)	107	82	111	111	74	61
	sigma	Uniform(100.0, 500.0)	409.80	369.46	332.04	493.91	490.70	261.11
BestAgentReminder	alpha	Uniform(0.3, 0.7)	0.67	0.42	0.34	0.45	0.42	0.53
	batch_size	RandInt(16, 256)	95	221	177	111	221	179
	learning_rate	LogUniform($10^{-4}, 10^{-3}$)	2.07e−4	2.58e−4	8.64e−4	3.12e−4	2.58e−4	1.25e−4
	sigma	Uniform(100.0, 500.0)	430.72	370.18	476.31	160.91	370.18	493.97
GA	mutation_rate	LogUniform($10^{-3}, 10^{-1}$)	3.96e−3	2.13e−3	1.86e−2	1.97e−2	4.91e−3	1.86e−2
	offspring_size	RandInt(50, 500)	160	426	230	245	130	230
	population_size	RandInt(50, 500)	217	397	409	444	496	409
Gflownet	learning_rate	LogUniform($10^{-5}, 10^{-3}$)	5.98e−5	1.72e−4	3.52e−4	1.17e−5	1.72e−4	3.33e−5
	momentum	Uniform(0.5, 0.9)	0.72	0.72	0.68	0.82	0.72	0.70
	sampling_tau	Uniform(0.8, 0.99)	0.87	0.96	0.83	0.97	0.96	0.93
GflownetDF	learning_rate	LogUniform($10^{-5}, 10^{-3}$)	8.25e−4	9.30e−4	1.37e−4	8.25e−4	3.75e−5	2.71e−5
	momentum	Uniform(0.5, 0.9)	0.83	0.78	0.60	0.83	0.77	0.57
	sampling_tau	Uniform(0.8, 0.99)	0.98	0.83	0.91	0.98	0.81	0.98
LSTM-HC	mols_to_sample	RandInt(8, 2048)	174	245	463	1695	859	1503
	optimize_n_epochs	RandInt(1, 6)	4	1	1	1	1	1
LSTM-PPO	batch_size	RandInt(64, 1024)	508	179	401	948	508	401
	clip_param	Uniform(0.1, 0.6)	0.11	0.19	0.41	0.12	0.22	0.41
	entropy_weight	Uniform(0.01, 1.0)	0.49	0.48	0.25	0.16	0.14	0.25
	episode_size	RandInt(64, 4096)	2277	1913	1121	2661	1932	1121
	kl_div_weight	RandInt(1, 10)	5	3	6	2	3	6
Mars	batch_size	RandInt(64, 512)	434	418	129	434	86	129
	n_layers	RandInt(1, 4)	1	3	2	1	1	2
	num_mols	RandInt(32, 512)	248	83	371	248	239	371
Mimosa	lamb	Uniform(0.1, 10.0)	7.07	5.84	2.03	7.07	5.84	1.73
	population_size	RandInt(50, 200)	150	199	87	150	199	85
	train_epoch	RandInt(1, 10)	2	2	9	2	2	8
Reinvent	batch_size	RandInt(256, 512)	260	288	417	462	454	464
	experience_replay	RandInt(0, 64)	49	34	54	38	12	50
	learning_rate	LogUniform($10^{-5}, 10^{-2}$)	1.00e−3	1.56e−3	3.25e−4	2.65e−4	2.24e−3	2.15e−4
	sigma	Uniform(100.0, 600.0)	582.10	540.86	296.22	357.38	244.20	283.52
SmilesGA	gene_size	RandInt(100, 600)	370	590	361	370	374	361
	n_mutations	RandInt(100, 300)	214	165	176	214	253	176
	population_size	RandInt(50, 200)	101	169	111	101	89	111
Stoned	generation_size	RandInt(50, 1000)	461	872	980	461	872	980

Table D1: Performance for the tested methods under a sample limit. Performance is given by the number of diverse hits (DivHits), novel diverse hits (NDivHits), and internal diversity (IntDiv). The internal diversity is calculated on the discovered hits.

	DRD2			GSK3β			JNK3		
	DivHits	NDivHits	IntDiv	DivHits	NDivHits	IntDiv	DivHits	NDivHits	IntDiv
AugMemory	$81^{\pm 19\%}$	$9^{\pm 75\%}$	$0.76^{\pm 0.01}$	$636^{\pm 6\%}$	$507^{\pm 12\%}$	$0.82^{\pm 0.00}$	$176^{\pm 11\%}$	$104^{\pm 13\%}$	$0.77^{\pm 0.00}$
AugmentedHC	$66^{\pm 11\%}$	$3^{\pm 44\%}$	$0.77^{\pm 0.01}$	$674^{\pm 11\%}$	$533^{\pm 11\%}$	$0.84^{\pm 0.00}$	$111^{\pm 27\%}$	$63^{\pm 41\%}$	$0.79^{\pm 0.01}$
LSTM-HC	$62^{\pm 16\%}$	$8^{\pm 37\%}$	$0.76^{\pm 0.01}$	$456^{\pm 9\%}$	$231^{\pm 16\%}$	$0.84^{\pm 0.01}$	$103^{\pm 13\%}$	$36^{\pm 17\%}$	$0.78^{\pm 0.00}$
BAR	$49^{\pm 11\%}$	$1^{\pm 56\%}$	$0.77^{\pm 0.02}$	$361^{\pm 8\%}$	$156^{\pm 11\%}$	$0.85^{\pm 0.00}$	$69^{\pm 20\%}$	$20^{\pm 19\%}$	$0.79^{\pm 0.00}$
Reinvent	$41^{\pm 25\%}$	$3^{\pm 53\%}$	$0.74^{\pm 0.02}$	$198^{\pm 18\%}$	$135^{\pm 24\%}$	$0.81^{\pm 0.01}$	$35^{\pm 11\%}$	$6^{\pm 68\%}$	$0.75^{\pm 0.01}$
GraphGA	$21^{\pm 31\%}$	$7^{\pm 55\%}$	$0.75^{\pm 0.04}$	$115^{\pm 14\%}$	$78^{\pm 15\%}$	$0.84^{\pm 0.00}$	$24^{\pm 37\%}$	$10^{\pm 61\%}$	$0.79^{\pm 0.01}$
VSRandom	$21^{\pm 12\%}$	$0^{\pm 0\%}$	$0.82^{\pm 0.01}$	$93^{\pm 6\%}$	$7^{\pm 12\%}$	$0.87^{\pm 0.00}$	$15^{\pm 13\%}$	$0^{\pm 0\%}$	$0.83^{\pm 0.01}$
LSTM-PPO	$14^{\pm 32\%}$	$0^{\pm 0\%}$	$0.81^{\pm 0.02}$	$108^{\pm 9\%}$	$16^{\pm 26\%}$	$0.87^{\pm 0.00}$	$13^{\pm 18\%}$	$1^{\pm 71\%}$	$0.81^{\pm 0.02}$
VSMaxMin	$19^{\pm 0\%}$	$0^{\pm 0\%}$	$0.88^{\pm 0.00}$	$68^{\pm 0\%}$	$8^{\pm 0\%}$	$0.89^{\pm 0.00}$	$9^{\pm 0\%}$	$0^{\pm 0\%}$	$0.88^{\pm 0.00}$
Mimosa	$6^{\pm 47\%}$	$0^{\pm 224\%}$	$0.80^{\pm 0.06}$	$23^{\pm 33\%}$	$8^{\pm 62\%}$	$0.84^{\pm 0.02}$	$8^{\pm 37\%}$	$3^{\pm 49\%}$	$0.78^{\pm 0.07}$
Mars	$3^{\pm 62\%}$	$0^{\pm 224\%}$	$0.39^{\pm 0.26}$	$39^{\pm 45\%}$	$36^{\pm 49\%}$	$0.81^{\pm 0.02}$	$4^{\pm 64\%}$	$2^{\pm 84\%}$	$0.61^{\pm 0.09}$
SmilesGA	$3^{\pm 80\%}$	$0^{\pm 224\%}$	$0.64^{\pm 0.36}$	$27^{\pm 12\%}$	$14^{\pm 26\%}$	$0.85^{\pm 0.01}$	$4^{\pm 83\%}$	$2^{\pm 87\%}$	$0.58^{\pm 0.34}$
GflownetDF	$0^{\pm 224\%}$	$0^{\pm 224\%}$	$0.00^{\pm 0.00}$	$77^{\pm 60\%}$	$73^{\pm 61\%}$	$0.81^{\pm 0.00}$	$0^{\pm 224\%}$	$0^{\pm 224\%}$	$0.00^{\pm 0.00}$
Stoned	$3^{\pm 41\%}$	$0^{\pm 0\%}$	$0.62^{\pm 0.15}$	$13^{\pm 19\%}$	$1^{\pm 64\%}$	$0.79^{\pm 0.06}$	$4^{\pm 37\%}$	$0^{\pm 224\%}$	$0.56^{\pm 0.15}$
Gflownet	$1^{\pm 122\%}$	$0^{\pm 0\%}$	$0.15^{\pm 0.33}$	$67^{\pm 75\%}$	$64^{\pm 77\%}$	$0.81^{\pm 0.01}$	$0^{\pm 0\%}$	$0^{\pm 0\%}$	$0.00^{\pm 0.00}$

Table D2: Performance for the tested methods under a time limit. Performance is given by the number of diverse hits (DivHits), novel diverse hits (NDivHits), and internal diversity (IntDiv). The internal diversity is calculated on the discovered hits.

	DRD2			GSK3β			JNK3		
	DivHits	NDivHits	IntDiv	DivHits	NDivHits	IntDiv	DivHits	NDivHits	IntDiv
LSTM-HC	$544^{\pm 7\%}$	$154^{\pm 15\%}$	$0.80^{\pm 0.00}$	$2620^{\pm 7\%}$	$2045^{\pm 8\%}$	$0.84^{\pm 0.00}$	$708^{\pm 13\%}$	$487^{\pm 13\%}$	$0.81^{\pm 0.00}$
AugmentedHC	$214^{\pm 31\%}$	$50^{\pm 42\%}$	$0.80^{\pm 0.01}$	$2543^{\pm 9\%}$	$2251^{\pm 10\%}$	$0.85^{\pm 0.00}$	$433^{\pm 14\%}$	$291^{\pm 15\%}$	$0.81^{\pm 0.00}$
Reinvent	$221^{\pm 5\%}$	$48^{\pm 16\%}$	$0.79^{\pm 0.01}$	$1315^{\pm 4\%}$	$1100^{\pm 5\%}$	$0.84^{\pm 0.00}$	$318^{\pm 11\%}$	$184^{\pm 17\%}$	$0.79^{\pm 0.01}$
BAR	$126^{\pm 4\%}$	$7^{\pm 30\%}$	$0.80^{\pm 0.01}$	$1469^{\pm 6\%}$	$1022^{\pm 10\%}$	$0.85^{\pm 0.00}$	$252^{\pm 7\%}$	$132^{\pm 15\%}$	$0.80^{\pm 0.01}$
VSMaxMin	$155^{\pm 0\%}$	$0^{\pm 0\%}$	$0.83^{\pm 0.00}$	$643^{\pm 0\%}$	$88^{\pm 0\%}$	$0.87^{\pm 0.00}$	$131^{\pm 0\%}$	$3^{\pm 0\%}$	$0.83^{\pm 0.00}$
AugMemory	$82^{\pm 11\%}$	$7^{\pm 44\%}$	$0.75^{\pm 0.01}$	$753^{\pm 9\%}$	$615^{\pm 12\%}$	$0.82^{\pm 0.01}$	$163^{\pm 26\%}$	$77^{\pm 33\%}$	$0.78^{\pm 0.01}$
GraphGA	$102^{\pm 27\%}$	$50^{\pm 21\%}$	$0.78^{\pm 0.01}$	$774^{\pm 10\%}$	$699^{\pm 11\%}$	$0.85^{\pm 0.00}$	$111^{\pm 56\%}$	$70^{\pm 68\%}$	$0.80^{\pm 0.01}$
VSRandom	$134^{\pm 3\%}$	$0^{\pm 0\%}$	$0.82^{\pm 0.00}$	$540^{\pm 2\%}$	$86^{\pm 4\%}$	$0.87^{\pm 0.00}$	$125^{\pm 4\%}$	$4^{\pm 12\%}$	$0.83^{\pm 0.00}$
LSTM-PPO	$39^{\pm 18\%}$	$0^{\pm 0\%}$	$0.81^{\pm 0.01}$	$308^{\pm 25\%}$	$69^{\pm 48\%}$	$0.87^{\pm 0.00}$	$30^{\pm 28\%}$	$2^{\pm 130\%}$	$0.82^{\pm 0.00}$
Mimosa	$5^{\pm 34\%}$	$0^{\pm 0\%}$	$0.82^{\pm 0.04}$	$26^{\pm 33\%}$	$8^{\pm 68\%}$	$0.84^{\pm 0.02}$	$6^{\pm 50\%}$	$1^{\pm 71\%}$	$0.74^{\pm 0.09}$
SmilesGA	$4^{\pm 35\%}$	$0^{\pm 224\%}$	$0.76^{\pm 0.11}$	$17^{\pm 17\%}$	$8^{\pm 33\%}$	$0.84^{\pm 0.04}$	$4^{\pm 59\%}$	$2^{\pm 82\%}$	$0.42^{\pm 0.40}$
Stoned	$3^{\pm 34\%}$	$0^{\pm 0\%}$	$0.54^{\pm 0.30}$	$15^{\pm 18\%}$	$1^{\pm 96\%}$	$0.76^{\pm 0.06}$	$4^{\pm 50\%}$	$0^{\pm 224\%}$	$0.61^{\pm 0.17}$
Gflownet	$0^{\pm 224\%}$	$0^{\pm 224\%}$	$0.00^{\pm 0.00}$	$112^{\pm 70\%}$	$108^{\pm 73\%}$	$0.80^{\pm 0.01}$	$0^{\pm 224\%}$	$0^{\pm 224\%}$	$0.00^{\pm 0.00}$
GflownetDF	$0^{\pm 137\%}$	$0^{\pm 224\%}$	$0.00^{\pm 0.00}$	$87^{\pm 53\%}$	$84^{\pm 51\%}$	$0.81^{\pm 0.01}$	$0^{\pm 224\%}$	$0^{\pm 224\%}$	$0.00^{\pm 0.00}$
Mars	$2^{\pm 100\%}$	$0^{\pm 0\%}$	$0.37^{\pm 0.34}$	$15^{\pm 36\%}$	$10^{\pm 45\%}$	$0.75^{\pm 0.03}$	$2^{\pm 122\%}$	$0^{\pm 224\%}$	$0.19^{\pm 0.19}$

D Extended results

D.1 Additional metrics

Tables D1 and D2 give extended results for both constraints settings, including the number of novel diverse hits and internal diversity of the found hits. The novel diverse hits are calculated as follows: First, we take the hits found by the generative model and from these remove all compounds that have a distance of less than 0.7 to any active compound in the training set. Then we calculate the #Circles metric on this reduced set. While the number of novel diverse hits is in general highly correlated with the number of diverse hits, the virtual screening methods generate relatively few novel diverse hits. This is because the virtual screening methods are biased toward finding compounds that are similar to the training set, which can be seen in similar distributions of the molecular properties of the generated molecules and the training set

Figure D1 shows the correlation between different diversity metrics, namely the number of hits, diverse hits, the number of novel diverse hits, and the internal diversity of the hits. We can see that the number of diverse hits is strongly correlated with the number of novel diverse hits. The internal diversity is only weakly correlated with these metrics. This means internal diversity values reported in previous studies are not a good indicator of the number of diverse (novel) hits found.

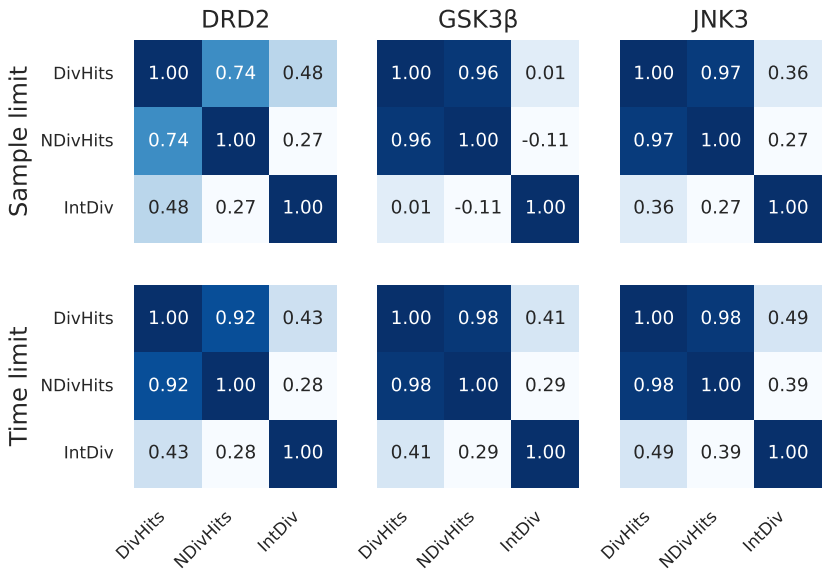

Figure D1: Correlation between different diversity metrics. The number of diverse hits is correlated with the number of novel diverse hits. The internal diversity is only weakly correlated with the other metrics.

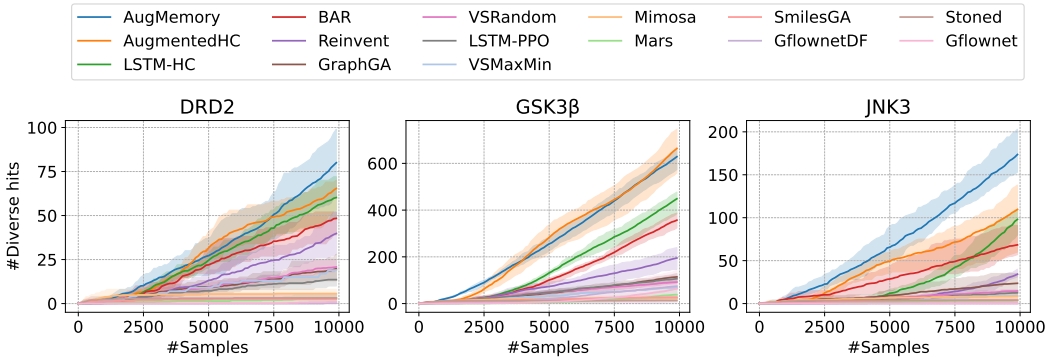

Figure D2: Number of diverse hits found by the tested methods over the number of scoring function evaluations.

D.2 Optimization curves

Figures D2 and D3 show the number of diverse hits found by the tested methods over the number of scoring function evaluations and the time elapsed. Most methods show no sign of saturating performance within the chosen limits. This shows that the comparison of the methods is only meaningful under standardized computing constraints. The method ranking remains somewhat constant throughout the optimization. Only single curves, like LSTM-HC on DRD2 in the sample-constrained setting, show a significant increase in performance towards the end of the optimization. Similarly, this holds for AugmentedHC in the time-constrained setting.

D.3 Molecular property distributions

In this section, we show distributions of the molecular weight (MW), the water-octanol partition coefficient (logP), the fraction of de-novo ECFP4 bits, the synthetic accessibility (SA), the quantitative estimate of drug-likeness (QED), and the length of the SMILES strings of the generated molecules. Figure D4 shows the distributions for the sample constrained setting, and Figure D5 shows the dis-

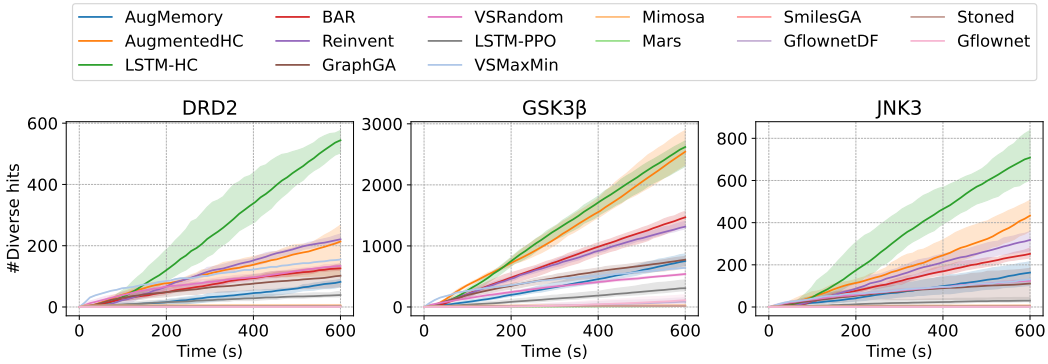

Figure D3: Number of diverse hits found by the tested methods over the elapsed time.

tributions for the time-constrained setting. For the properties used in the property filter, also show the used thresholds.

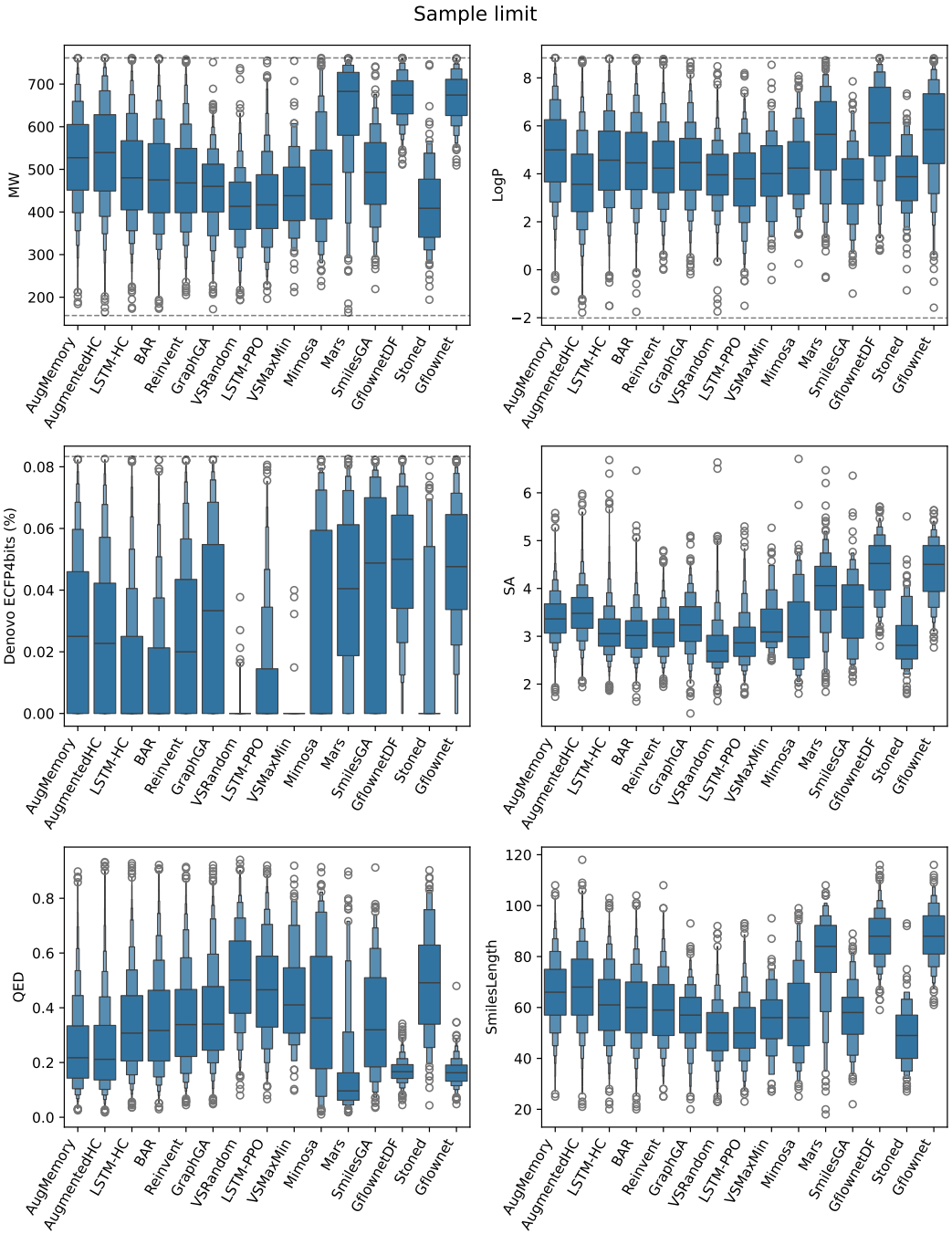

Figure D4: Distributions of the molecular properties of the generated molecules for the sample-constrained setting. The dashed lines indicate the thresholds used in the property filter.
Results are aggregated over the three optimization tasks.

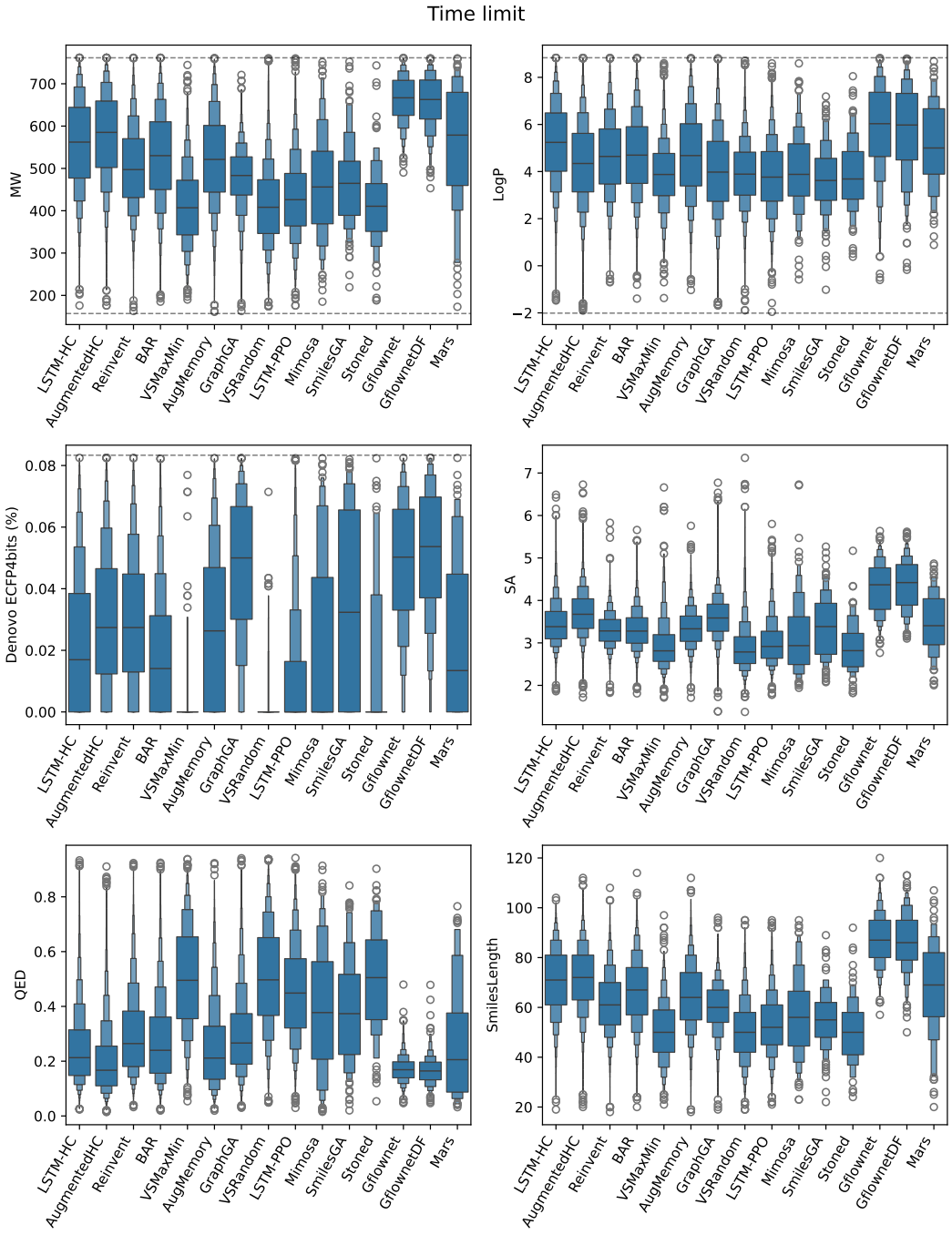

Figure D5: Distributions of the molecular properties of the generated molecules for the time-constrained setting. The dashed lines indicate the thresholds used in the property filter.
Results are aggregated over the three optimization tasks.