A Fresh Look at *De Novo* Molecular Design Benchmarks

Anonymous Author(s)
Affiliation
Address
email

Abstract

De novo molecular design is a thriving research area in machine learning (ML) that lacks ubiquitous, high-quality, standardized benchmark tasks. Many existing benchmark tasks do not precisely specify a training dataset or an evaluation budget, which is problematic as they can significantly affect the performance of ML algorithms. This work elucidates the effect of dataset sizes and experimental budgets on established molecular optimization methods through a comprehensive evaluation with 11 selected benchmark tasks. We observe that the dataset size and budget significantly impact all methods' performance and relative ranking, suggesting that a meaningful comparison requires more than a single benchmark setup. Our results also highlight the relative difficulty of benchmarks, implying in particular that logP and QED are poor objectives. We end by offering guidance to researchers on their choice of experiments.

1 Introduction

In recent years, *de novo* molecular design has seen increased attention from the ML community. The goal is to produce data-driven algorithms that can effectively and efficiently guide the design of novel molecules and materials with desirable properties such as drug efficacy. Mathematically the problem is typically posed as the optimization of a black-box *objective function* over molecular space. For real-world problems, evaluating these objective functions typically involves costly and time-consuming computational or wet-lab experiments; for this reason, researchers often assess their algorithms on fast and approximate computational objectives instead. Currently, the community still lacks ubiquitous benchmarks like Imagenet [Deng et al., 2009], and many nascent benchmarks suffer from severe limitations, such as

1. **Unrealistic Optimization Objectives**: Simply maximizing molecular properties such as the logP, a metric for the solubility of a compound in non-polar solvents, or the "Quantitative Estimate of Druglikeness", or QED, have limited medicinal value. Further, such objectives are more manageable than real-world *de novo* molecular design tasks and can be easily solved with basic random-search-like algorithms (see Brown et al. [2019] and section 3).
2. **Lack of Canonical Datasets**: The size and composition of the training set is known to significantly impact the performance of virtually every ML algorithm. Despite this, many *de novo* design benchmark tasks either lack an accompanying training set, or are frequently used with non-standard, modified training sets (e.g. subsampling, adding unlabelled data).
3. **Unspecified Evaluation Budgets**: Real *de novo* design problems that require experimental evaluation will be constrained by a limited time and resource budget, which is not reflected in most *de novo* design benchmarks. Researchers either focus on the setting in which there is an infinite budget or set an arbitrary finite budget. This is problematic since the available

Submitted to NeurIPS 2021 AI for Science Workshop.

number of objective function evaluations can fundamentally change the nature and difficulty of an optimization problem. For example, a small budget may favor exploitative algorithms, whereas a larger budget requires a balance of exploration and exploitation.

The field of *de novo* design would benefit from the adoption of high-quality standardized benchmarks. These benchmarks should include an explicitly defined objective function, dataset, and evaluation budget. Furthermore, they ought to closely reflect the difficulty of *de novo* design problems. However, the *best* benchmark is *a priori* unclear partly because there are no known systematic evaluations of benchmark candidates. This paper provides a preliminary comparison of different benchmark candidates by running a few established algorithms on many previously proposed benchmarks. Critically, unlike previous work, we test *16 variations* of each benchmark with different training set sizes and evaluation budgets rather than choosing a single setting. In total, we performed over 200 experiments. We analyze the difficulty of each benchmark in detail and provide fresh insights into the merits of each objective function, dataset, and budget combination. Based on our results, we offer concrete suggestions to researchers and propose directions for the field to adopt better benchmarks: (1) fix datasets and budgets for a fair comparison, (2) test multiple datasets and budget settings for each objective, (3) do not use (penalized) logP and QED objectives, (4) use GuacaMol and DOCKSTRING instead.

2 Experimental Setup

Starting from a list of algorithms, objective functions, datasets, and budgets, we ran an experiment with every algorithm/objective/dataset/budget combination. We performed three replicates of each experiment to account for randomness. Below, we specify the experimental setups.

***De novo* design algorithms.** In total, we tested five different algorithms: "Dataset Best", "Random ZINC", Graph GA, SELFIES GA, and GP BO. "Dataset Best" is a trivial algorithm that performs no queries and returns the best molecule in the training set. "Random ZINC" ignores the training set and uses the budget to randomly evaluate molecules from the ZINC dataset of ~1 billion purchasable drug-like molecules [Irwin et al., 2012, 2020]. We chose these trivial baselines as a sanity check: if either of these algorithms is competitive, the task must be too easy. Graph GA and SELFIES GA are genetic algorithms (GAs) that use graph-based [Jensen, 2019] and string-based [Krenn et al., 2020] representations of molecules, respectively. We selected these algorithms to represent prototypical exploratory algorithms: if these algorithms are competitive, the task can be solved chiefly via exploration. Gaussian process Bayesian optimization (GP BO) [Srinivas et al., 2010] is a well-established algorithm known for its capacity for both exploration and exploitation and its good performance in the low-budget regime. *A priori* GP BO was expected to be the best-performing algorithm. Further details on these algorithms are given in Appendix B. We are aware that many algorithm classes and variants are not represented here: these algorithms were chosen because they are well-established, fast, work well with a wide range of dataset sizes and budgets, and have very few hyperparameters to tune.

Objective functions. We examined 11 objective functions in total. Maximization of logP [Wildman and Crippen, 1999], penalized logP [Gómez-Bombarelli et al., 2018], and QED [Bickerton et al., 2012] were chosen due to their frequent use in previous work, despite their well-documented limitations. We selected the Celecoxib and Troglitazone rediscovery, Median Molecules 1 and 2, and Osimertinib and Zaleplon MPO (*multiproperty objective*) objectives from the 20 goal-directed generation objectives in the GuacaMol benchmark suite [Brown et al., 2019] (chosen to represent the three major task types in their benchmark suite). Finally, we selected two tasks from the recently published DOCKSTRING study [García Ortegón et al., 2021]: F2 and selective JAK2. These tasks both involve proposing drug-like molecules with low binding free energies to a target protein.

Datasets. In this work, we vary only the size of the dataset and not its distribution. We choose to examine datasets of size $\{10^2, 10^3, 10^4, 10^5\}$ produced by uniformly subsampling molecules from a larger dataset (without replacement). For logP, QED, and GuacaMol tasks, the GuacaMol dataset consisting of ~10^6 molecules is employed. For the DOCKSTRING tasks, ~250 000 molecules from the accompanying dataset are used.

Evaluation Budget. For each experiment, we fixed the number of objective function evaluations, i.e., computational budget. The list of budgets was: $\{10^2, 10^3, 10^4, 10^5\}$. Due to its relatively high computational cost, we omit the 10^5 budget for the DOCKSTRING tasks.

3 Results

The results are summarized in Figure 1 (see also Table A.1). First, it can be seen that the dataset size and budget affect the performance of all algorithms. As one would expect, algorithms generally tend to achieve higher scores with larger datasets and higher budgets. But there are exceptions: for example, the performance of GP BO with a budget of 10^5 is virtually same for all dataset sizes on Celecoxib. Second, for each objective, the ranking of different algorithms can change depending on the budget and initial dataset size. For example, for p-logP with an initial dataset size and budget of 10^5, the order is SELFIES GA > Graph GA > GP BO, while with an initial dataset size and budget 10^3 the order is reversed. Third, for a fixed initial dataset and budget, the order can change for different objectives: for example, for an initial dataset size and budget of 10^5 the order on Zaleplon MPO is GP BO > Graph GA > SELFIES GA while it is Graph GA > SELFIES GA > GP BO on Osimertinib MPO. Overall our findings indicate that the selected tasks vary significantly in difficulty and that measuring performance on one task is not a reliable predictor of performance on another one. We give a brief interpretation of the results for each objective below, with more details in Appendix A.

- **logP and penalized logP** are exploited by algorithms that propose large molecules (such as SELFIES GA). We recommend *against* using these objectives.
- **QED** is too easy: performance saturates at 0.948 with even a moderate budget and is also achieved by all baseline algorithms, *including* random ZINC and dataset best.
- **Celecoxib and Troglitazone Rediscovery** are such that some but not all algorithms achieve the maximum score of 1.0 with a high budget or large initial dataset. Troglitazone is the more challenging of the two since the perfect score is achieved less often.
- **Median Molecules 1 and 2** showcase that the performance of algorithms varies significantly with dataset size and budget. Median mols 1 seems more sensitive to dataset size and budget.
- **Osimertinib and Zaleplon MPO** are difficult and look qualitatively different: Osimertinib MPO seems to require significant exploration, whereas Zaleplon MPO seems to require both significant exploration and exploitation.
- **DOCKSTRING tasks** seem more exploratory as performance depends weakly on dataset size, but the strong performance of GP BO suggests that exploration alone is insufficient. Overall they appear to be qualitatively different from all other Guacamol objectives.

4 Conclusions

Based on the results, we have the following recommendations:

1. **Fix datasets and budgets for a fair comparison.** The performance of the baseline algorithms varies considerably for the same objective with different budgets/dataset sizes. Therefore, to fairly compare algorithms, the same dataset and budget must be used.
2. **Test multiple datasets and budget settings for each objective.** The relative performance of algorithms can depend on the initial dataset and budget. This dependence can be easily missed if only one setting for the dataset and budget is used. If it is necessary to choose only one configuration, a principled justification should be given.
3. **Do *not* use (penalized) logP and QED objectives.** Despite previous works decrying how poor these objectives are [Brown et al., 2019, García Ortegón et al., 2021], they continue to be used. Our results provide further evidence that these tasks are poor benchmarks, and we strongly discourage their use.
4. **Use GuacaMol and DOCKSTRING instead.** The GuacaMol benchmarks are less trivial and have objectives of varying difficulty, with the MPO objectives being particularly challenging. The DOCKSTRING objectives are difficult and qualitatively different from Guacamol tasks. They have the bonus of being medicinally relevant, albeit computationally expensive.

Ultimately this paper is only a single step in a long journey towards building better benchmarks. Our main contribution has been to highlight the importance of specifying the dataset and budget of a benchmark, and we hope that the field moves towards standardizing and specifying these in their future work. We hope this inspires the community to re-access the benchmarks that they are using.

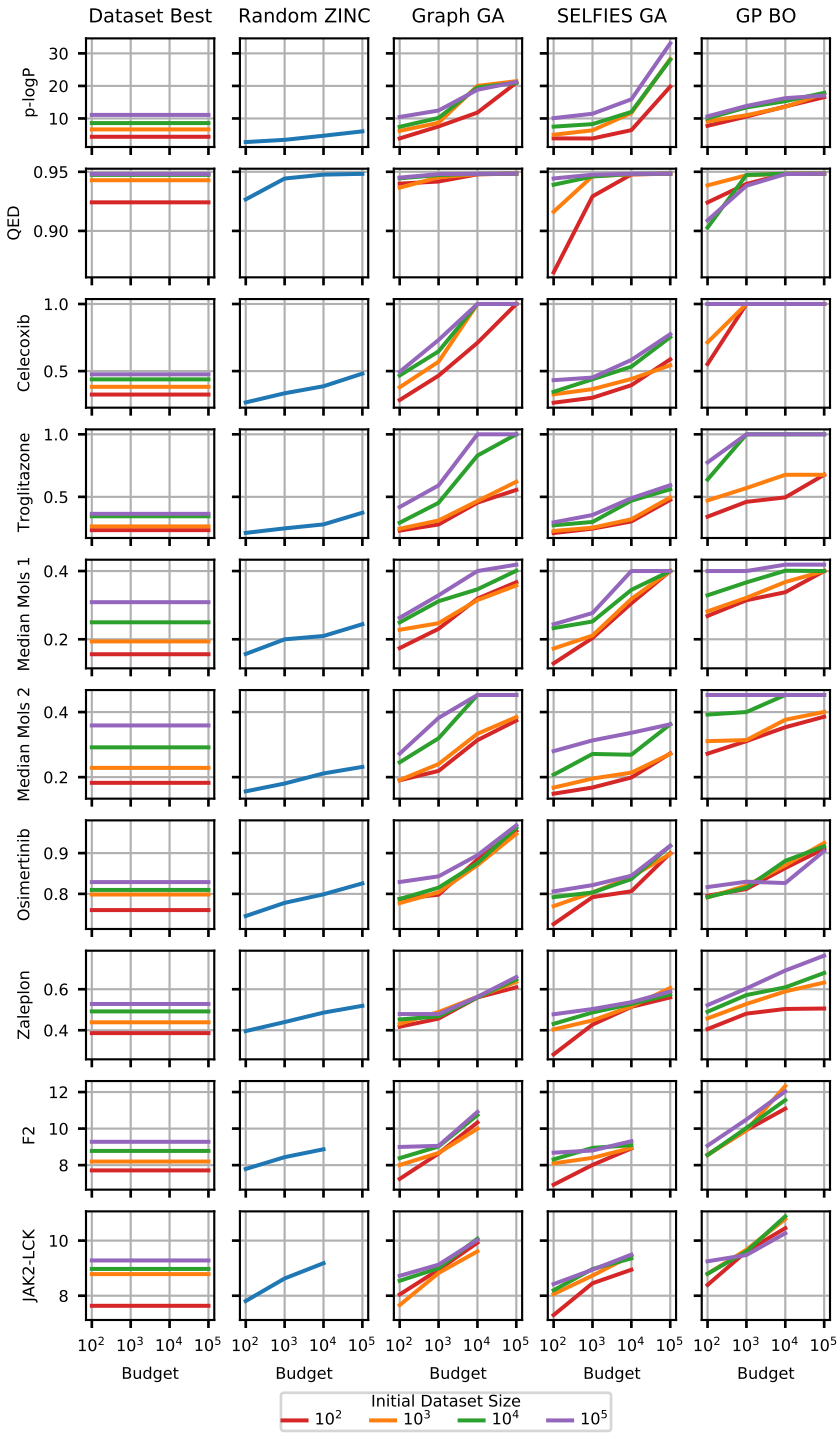

Figure 1: Task scores as a function of computational budged achieved by five methods for varying initial dataset sizes. For the tasks F2 and JAK2-LCK, not all experiments could be performed due to limited computational resources.

References

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

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.*, 59(3):1096–1108, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Miguel García Ortegón, Gregor NC Simm, Austin Tripp, José Miguel Hernández Lobato, Andreas Bender, and Sergio Bacallado de Lara. Dockstring: easy molecular docking yields better benchmarks for ligand design. *submitted*, 2021.

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 Cent. Sci.*, 4(2):268–276, 2018.

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*, 2017.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *J. Chem. Inf. Model.*, 52(7):1757–1768, 2012.

John J Irwin, Khanh G Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R Wong, Munkhzul Khurelbaatar, Yurii S Moroz, John Mayfield, and Roger A Sayle. Zinc20—a free ultralarge-scale chemical database for ligand discovery. *J. Chem. Inf. Model.*, 60(12):6065–6073, 2020.

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, 2019.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.*, 1(4):045024, 2020.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in Neural Information Processing Systems*, 31:7795–7804, 2018.

AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. *Chem. Sci.*, 2021.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022, 2010.

Taffee T Tanimoto. Elementary mathematical theory of classification and prediction. *IBM Internal Report*, 1958.

Scott A Wildman and Gordon M Crippen. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.*, 39(5):868–873, 1999.

Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6412–6422, 2018.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Sci. Rep.*, 9(1):1–10, 2019.

A Additional Results

The complete results are in Table A.1. We give a more detailed assessment of the objectives tested below:

logP and penalized logP. With a sufficient budget, all non-trivial algorithms significantly outperform the best in the dataset. logP and penalized logP are easy to maximize by adding more (carbon) atoms to a molecule. Therefore, this objective can efficiently be maximized *ad infinitum* by algorithms. Graph GA, SELFIES GA, and GP BO do this, and consequently, we see steadily increasing performance with an increasing budget. Graph GA and GP BO have an internal bias against proposing molecules with high molecular weight. Therefore, they are outperformed by SELFIES GA, which does not have such an internal bias. However, SELFIES GA performs worse at most other tasks that are more demanding. In conclusion, logP variants are poor proxies for different tasks, and we recommend against their use.

QED. QED appears to have a global maximum at 0.948, as none of our algorithms and previous works failed to find a molecule with a higher score [Guimaraes et al., 2017, Liu et al., 2018, You et al., 2018, Jin et al., 2018, Zhou et al., 2019]. All algorithms reached this value with a high budget. The trivial Dataset Best and Random ZINC baselines even with moderate budgets. This suggests that the task is too easy to determine the relative performance of algorithms; at best, it could be used as a sanity check or as a benchmark in the extremely low-budget setting. Overall we recommend against using this objective.

Celecoxib and Troglitazone Rediscovery. Both of these objectives are challenging, as the trivial baselines and SELFIES GA do not achieve the theoretical maximum of 1.0. However, both Graph GA and GP BO obtain good results with a moderate budget and a large dataset. Celecoxib seems to be slightly easier than Troglitazone as the highest possible score is achieved with multiple settings. These objectives could be employed in the low data or low budget setting or as a toy objective.

Median Molecules 1 and 2. The performance of all algorithms varies visibly with dataset size and budget. Median Molecules 1 seems more difficult as good performance is achieved only with a high budget and a large dataset. In contrast, the performance on the Median Molecules 2 task can be improved with just a large dataset (at least with GP BO). For small budgets the best molecule in the dataset is competitive.

Osimertinib and Zaleplon MPO. The plots for Osimertinib MPO resemble those of logP, with performance depending more strongly on budget than on dataset size. This suggests that the task requires more exploration than exploitation and that the GuacaMol dataset does not contain many molecules close to the optimum. The fact that both genetic algorithms outperform GP BO bolsters this claim since these algorithms are more exploratory. For Zaleplon MPO, the performance of GP BO increases significantly with dataset size, whereas for the other methods there is comparatively little increase. This behaviour is not seen with any other objective function, suggesting that Zaleplon MPO has unique difficulties not present in the other benchmarks. GP BO also significantly outperforms the genetic algorithms, suggesting that exploitation is necessary to achieve competitive results. Since GP BO achieved the best score in the maximum data and budget setting, we conclude that Zaleplon MPO requires significant exploration and exploitation. This appears to be the most challenging GuacaMol task out of the ones surveyed.

DOCKSTRING tasks. First, all algorithms see increased performance with an increasing budget, suggesting that the global maximum is not in or near the dataset. Secondly, performance varies less significantly with dataset size than other tasks, suggesting that the dataset does not contain many points near the optimum. Further, significant exploration is required for good performance. Unlike Osimertinib MPO, GP BO significantly outperforms both genetic algorithms with a high budget, showing that exploitation is necessary (or that random exploration is less likely to succeed). Also unlike the Guacamol tasks, SELFIES GA fails to outperform the best in the dataset in almost every setting. These tasks are qualitatively different from any other GuacaMol tasks and would be good benchmarks for novel algorithms in various settings, mainly owing to their medicinal relevance.

Table A.1: Median best molecule across three runs for each task. A subset of this data are visualized in Figure 1. The abbreviated task names are penalized logP (p-logP), Celecoxib rediscovery (Cel.), Troglitazone rediscovery (Trog.), Median Molecules 1 and 2 (MM1 and MM2), Osimertinib MPO (O MPO), Zaleplon MPO (Z MPO). F2 and JAK2-LCK correspond to the F2 and selective JAK2 task from DOCKSTRING.

Method	Initial Dataset Size	Budget	logP	p-logP	QED	Cel.	Trog.	MM1	MM2	O MPO	Z MPO	F2	JAK2-LCK
Dataset Best	10^2	0	8.359	4.374	0.924	0.325	0.234	0.156	0.183	0.760	0.386	7.712	7.634
	10^3	0	11.703	6.633	0.943	0.383	0.264	0.194	0.229	0.798	0.439	8.197	8.784
	10^4	0	15.039	8.581	0.947	0.438	0.345	0.250	0.292	0.810	0.492	8.778	8.972
	10^5	0	19.775	11.095	0.948	0.476	0.364	0.309	0.359	0.829	0.528	9.278	9.280
Random ZINC	0	10^2	5.131	2.751	0.927	0.267	0.212	0.157	0.156	0.746	0.396	7.789	7.808
		10^3	6.152	3.432	0.944	0.335	0.248	0.200	0.180	0.778	0.440	8.438	8.626
		10^4	7.772	4.692	0.948	0.387	0.280	0.210	0.211	0.799	0.486	8.867	9.180
		10^5	11.132	6.038	0.948	0.481	0.373	0.244	0.232	0.826	0.519	-	-
Graph GA	10^2	10^2	9.973	3.871	0.940	0.285	0.230	0.174	0.190	0.787	0.417	7.246	8.044
		10^3	12.564	7.542	0.942	0.466	0.279	0.231	0.219	0.798	0.457	8.637	8.942
		10^4	21.121	11.810	0.948	0.711	0.454	0.319	0.314	0.884	0.560	10.336	9.923
		10^5	30.864	20.980	0.948	1.000	0.556	0.367	0.374	0.954	0.609	-	-
	10^3	10^2	10.527	6.182	0.937	0.379	0.245	0.228	0.191	0.777	0.431	8.006	7.660
		10^3	14.768	8.676	0.945	0.570	0.311	0.247	0.240	0.804	0.489	8.665	8.823
		10^4	26.910	20.016	0.948	1.000	0.466	0.315	0.333	0.870	0.563	9.992	9.609
		10^5	31.396	21.457	0.948	1.000	0.621	0.358	0.385	0.947	0.636	-	-
	10^4	10^2	10.643	7.443	0.944	0.467	0.293	0.249	0.246	0.788	0.453	8.382	8.542
		10^3	17.172	10.174	0.947	0.647	0.452	0.312	0.319	0.815	0.468	9.019	8.999
		10^4	27.495	19.539	0.948	1.000	0.830	0.346	0.453	0.875	0.560	10.742	10.069
		10^5	32.502	20.920	0.948	1.000	1.000	0.401	0.453	0.962	0.648	-	-
	10^5	10^2	15.000	10.458	0.945	0.495	0.419	0.263	0.272	0.829	0.479	8.995	8.722
		10^3	19.182	12.444	0.948	0.732	0.591	0.329	0.382	0.843	0.479	9.056	9.127
		10^4	27.271	18.820	0.948	1.000	1.000	0.400	0.453	0.895	0.563	10.911	10.023
		10^5	32.957	21.206	0.948	1.000	1.000	0.419	0.453	0.968	0.659	-	-
SELFIES GA	10^2	10^2	6.740	3.877	0.865	0.264	0.211	0.129	0.149	0.726	0.282	6.928	7.299
		10^3	9.886	3.849	0.929	0.301	0.246	0.204	0.168	0.792	0.427	8.011	8.452
		10^4	22.583	6.411	0.948	0.395	0.303	0.306	0.199	0.806	0.514	8.914	8.943
		10^5	81.522	19.784	0.948	0.588	0.476	0.400	0.273	0.900	0.560	-	-
	10^3	10^2	9.672	5.008	0.916	0.327	0.227	0.173	0.168	0.770	0.404	8.096	8.059
		10^3	13.813	6.359	0.946	0.366	0.253	0.211	0.196	0.803	0.448	8.399	8.731
		10^4	25.450	11.679	0.948	0.441	0.321	0.318	0.214	0.842	0.516	8.942	9.453
		10^5	78.788	28.247	0.948	0.543	0.495	0.400	0.272	0.898	0.606	-	-
	10^4	10^2	11.668	7.490	0.939	0.346	0.273	0.233	0.207	0.792	0.431	8.313	8.198
		10^3	17.249	8.262	0.946	0.439	0.299	0.252	0.271	0.803	0.487	8.944	8.969
		10^4	25.655	12.017	0.948	0.533	0.469	0.345	0.269	0.836	0.528	9.092	9.354
		10^5	77.323	28.065	0.948	0.753	0.560	0.400	0.362	0.918	0.574	-	-
	10^5	10^2	15.811	10.094	0.944	0.432	0.296	0.244	0.280	0.806	0.478	8.682	8.423
		10^3	22.705	11.457	0.948	0.452	0.355	0.277	0.313	0.821	0.503	8.800	8.942
		10^4	31.087	15.885	0.948	0.583	0.489	0.400	0.336	0.844	0.537	9.311	9.490
		10^5	78.489	33.089	0.948	0.775	0.593	0.400	0.362	0.918	0.589	-	-
GP BO	10^2	10^2	15.371	7.737	0.924	0.553	0.341	0.268	0.272	0.795	0.406	8.569	8.391
		10^3	18.521	10.593	0.940	1.000	0.460	0.314	0.311	0.812	0.481	9.917	9.611
		10^4	25.532	13.689	0.948	1.000	0.496	0.338	0.353	0.863	0.504	11.091	10.451
		10^5	35.561	16.546	0.948	1.000	0.676	0.400	0.386	0.911	0.506	-	-
	10^3	10^2	18.238	9.124	0.939	0.714	0.471	0.282	0.311	0.791	0.458	8.563	8.794
		10^3	24.179	10.991	0.947	1.000	0.570	0.322	0.314	0.820	0.528	9.907	9.668
		10^4	27.543	13.569	0.948	1.000	0.676	0.368	0.376	0.870	0.589	12.332	10.783
		10^5	31.021	17.705	0.948	1.000	0.676	0.400	0.400	0.925	0.632	-	-
	10^4	10^2	20.185	10.068	0.903	1.000	0.638	0.329	0.392	0.792	0.490	8.551	8.795
		10^3	23.701	13.394	0.947	1.000	1.000	0.367	0.400	0.815	0.572	10.036	9.606
		10^4	28.135	15.311	0.948	1.000	1.000	0.401	0.453	0.881	0.609	11.559	10.881
		10^5	30.595	17.873	0.948	1.000	1.000	0.400	0.453	0.917	0.679	-	-
	10^5	10^2	19.829	10.650	0.909	1.000	0.776	0.400	0.453	0.817	0.522	9.066	9.249
		10^3	25.986	13.841	0.938	1.000	1.000	0.400	0.453	0.830	0.604	10.497	9.476
		10^4	27.749	16.246	0.948	1.000	1.000	0.419	0.453	0.827	0.691	12.037	10.266
		10^5	30.742	17.042	0.948	1.000	1.000	0.419	0.453	0.905	0.764	-	-

B Experimental Details

The implementation of Graph GA was taken directly from the García Ortegón et al. [2021]. The implementation of GP BO was a basic Bayesian optimization loop using a GP with the Tanimoto kernel [Tanimoto, 1958] and an upper confidence bound acquisition function [Srinivas et al., 2010]. The implementation was based on that of García Ortegón et al. [2021], but was modified to use a larger batch size for the budget of 10^5. Further, it was modified to employ a subset of the data at all times rather than simply starting with a subset of the data. The SELFIES GA implementation was

based heavily on the Graph GA, except that the mutation step used the SELFIES [Krenn et al., 2020] genetic algorithm mutation function from Nigam et al. [2021].

All code will be made available on github at a later date.