OptiMol : Optimization of binding affinities in chemical space for drug discovery

A Preprint

Jacques Boitreaud *
Department of Computer Science
McGill University
jacques.boitreaud@mail.mcgill.ca

Vincent Mallet *
Pasteur Institute
Les Mines-Paristech
vincent.mallet96@gmail.com

Carlos Oliver
Department of Computer Science, McGill University
Montreal Institute for Learning Algorithms
carlos.gonzalezoliver@mail.mcgill.ca

Jerome Waldispühl
Department of Computer Science
McGill University
jerome.waldispuhl@mail.mcgill.ca

Abstract

Ligand-based drug design has recently benefited from the boost of deep generative models. These models enable extensive explorations of the chemical space, and provide a platform for molecular optimization. However, current state of the art methods do not leverage the structure of the target, which is known to play a key role in the interaction. We propose an optimization pipeline that leverages complementary structure-based and ligand-based methods. Instead of performing docking on a fixed drug bank, we iteratively select promising compounds in the whole chemical space using a ligand-centered generative model. Molecular docking is then used as an oracle to guide compound optimization. This allows to iteratively generate leads that better fit the target structure, in a closed optimization loop, without prior knowledge about bio-actives. For this purpose, we introduce a new graph to selfies VAE which is competitive with the state of the art and benefits from faster decoding than graph to graph methods. We also successfully optimize the generation of molecules towards high docking scores, enabling a ten-fold augmentation of high-scoring compounds found with a fixed computational budget.
Availability: Code is available on GitHub
Contact: jeromew@mcgill.ca

1 Introduction

Molecular optimization, also known as inverse design, consists in designing compounds that have desired drug-like properties and biological activity. In practice, this often consists in maximizing a compound's affinity to a given target. In the structure-based approach, an ensemble of lead candidates is obtained by screening fixed libraries. Molecular docking methods estimate the interaction energy of the each compound with the target. This step is computationally demanding and limits the screening to $\sim 10^6 - 10^9$ compounds compared to the drug-like chemical space that contains an estimated $\sim 10^{60}$ compounds. In the ligand-based approach, known bio-actives are used to induce structure-activity relationships, and drug-like leads with similar structures are proposed. This approach only implicitly leverages the target structure, and inherently limits the diversity of generated molecules. Therefore, it is not well suited for finding compounds with new binding modes to the target. It also relies on the assumption that molecules with similar structure

[*] Both authors contributed equally

are likely to exhibit similar bio-activity, which does not hold true in the whole chemical space, especially near activity cliffs [1].

Active learning and automated synthesis have been identified as promising research directions to accelerate the drug discovery process [2]. By allowing a model to iteratively query the most informative data, and learn from experimental answers to these queries, these methods enable a guided and data-efficient exploration of the activity landscape in the chemical space. Inspired by such closed-loop strategies, we propose to iteratively search the chemical space for promising leads, guided by a structure-based assessment of their activity. By combining molecular docking and a latent-variable generative model, our framework finds regions of high affinity to a target in the chemical space.

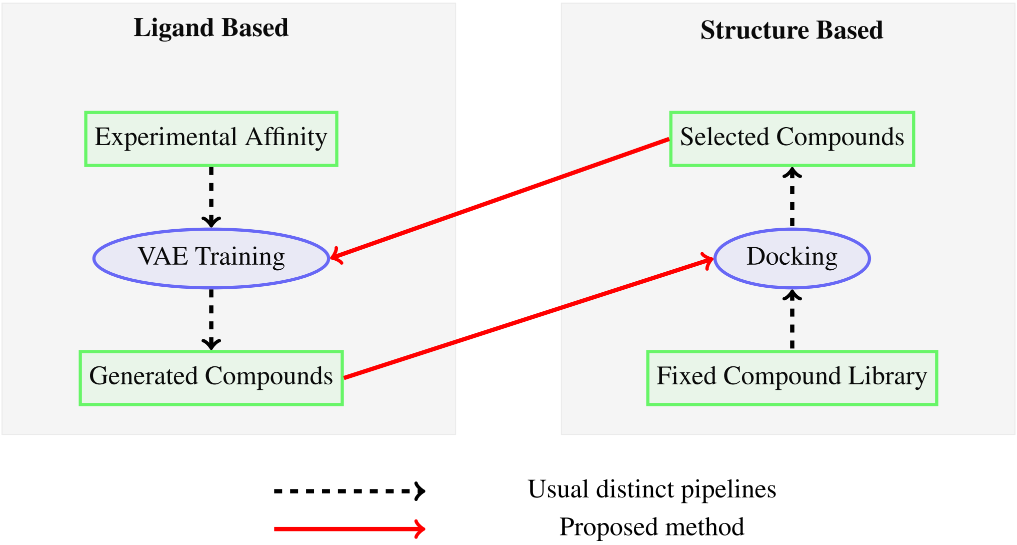

Figure 1: Closing the loop : In ligand-based approaches, a generative model is biased with experimental affinities to produce more active compounds. In structure based approaches, a fixed library is screened for actives. We propose to dock the compounds produced by the generative model, and to use the results of the docking to fine-tune the ligand-based generative model

2 Related work

2.1 Molecule representation

Molecules can be represented in several ways, trading off between the accuracy of the depiction and its computational advantages. In decreasing order of richness of representation, molecules are represented as ensembles of 3D, static 3D, molecular graphs, SMILES, Selfies [3] or fingerprints. Each of these representation has drawbacks. Using 3D objects with no preferred orientation in machine learning pipelines is under research with promising results [4, 5] but not yet established. Molecular graphs are efficient to encode information but deconvolution and generation is harder [6, 7]. Finally string representations were used first as they benefited from advances in natural language processing. SMILES were extensively used but when generating SMILES, some sequences are invalid. This problem was recently solved by the introduction of Selfies [3] that are all valid sequences, but are sensitive to small modifications.

The choice of the representation of molecules has a direct impact on the optimization process : The discrete nature of sequences and graphs does not allow for continuous optimization. This can be addressed by considering the sampling of the chemical space as a sequential process that can be formulated as a reinforcement learning problem [8]. Another approach uses Variational Autoencoders (VAE) [9]. These methods learn a continuous chemical space by reconstructing or translating these representations [3, 6, 10, 11, 12, 13, 14]. The latent space can be interpreted as a flattened manifold, of which the structure and geometry mostly reflect chemical similarity. Turning the chemical space into a euclidean space enables to continuously navigate in the space, which opens the door to classical optimization methods.

2.2 Molecular optimization

Direct molecular optimization was first introduced in [10]. Subsequently, a string of papers addressed this problem with a variety of approaches as explained in reviews [15, 16]. Formally we are looking for compounds $\mathbf{x}$ that maximize a function $\mathbf{f}(\mathbf{x})$. This function is often a chemical property that makes a compound more drug-like such as QED or solubility, or a more complex property, such as bio-activity. Several properties of $\mathbf{f}$ affect its optimization: differentiability, dimension of the output space , evaluation cost and smoothness. Molecular optimization can be further subdivided into lead discovery, which corresponds to this setting, and lead optimization, where we start from a given seed compound, meaning $\mathbf{x}$ is constrained to a region of the chemical space.

The size of the chemical space and the difficulty to accurately estimate the objective $\mathbf{f}$ without *in-vitro* tests makes it unreasonable to look for just one candidate. Moreover, $\mathbf{f}$ is often a simple surrogate for a complex, phenotypical endpoint. Hence the current approach to molecular optimization is to search for ensembles of compounds with enhanced estimated properties and then conduct in-vitro tests. Formally, we are not after the global maximum of $\mathbf{f}$. We rather wish to find a non-trivial distribution q from which we can sample and that maximizes $\mathbb{E}_{\mathbf{x}\sim q}(\mathbf{f}(\mathbf{x}))$. The exact solution would be a trivial distribution always returning the global maxima. To avoid this, we could simultaneously maximize the distribution entropy but in practice, the distribution found is sub-optimal, making it non trivial.

Guided generation using Reinforcement Learning By decomposing the molecule generation as a sequence of actions, we can build a generative model using a probabilistic RL agent. Guided generation consists in biasing the generation towards compounds that optimize $\mathbf{f}$, easily including non differentiable objectives [8, 17]. An adversarial term can be added to ensure that we keep generating realistic compounds [18, 19]. One caveat in using these methods is that for multidimensional objective functions, the rewards get sparse and the training gets harder [18].

Latent space optimization Once molecules are embedded in a continuous space, classical optimization schemes become possible. The latent space geometry is induced by the chosen representation, and often reflects chemical similarity. Bayesian optimization was used first [6, 10, 20, 21], followed by constrained Bayesian optimization [22, 23] and swarm optimization [2]. Another approach is to approximate $\mathbf{f}$ with a differentiable function and to conduct gradient ascent in the latent space. A potential advantage of this method is that the function mapping from latent space to $\mathbf{f}$ can be learned jointly with the generative model, contributing to shape the latent space [12].

Molecular translation Matching Molecular Pairs Analysis were recently used by [24] to train a translational VAE that translates a molecule into a similar one with better target property: the translational model learns to take an optimization step, given a starting point. In [25], the authors show that molecules can be optimized for a target property by recursively taking such steps in the chemical space. This step-wise approach is limited to lead optimization, but seems well suited to handle activity cliffs.

Miscellaneous The authors of [26] use a fragment based approach where they automatically learn a library of activity inducing fragments and then generate compounds combining them. [27] train a Generative Adversarial Network (GAN) to mimic the active compounds distribution in the latent space. This requires the distribution of actives to be fixed beforehand. Another approach is to use the same kind of strategy in an iterated way [28], fine-tuning a generative model on its most successful outputs. Other approaches of this kind with better statistical grounding exist [29, 30], but we are unaware of their application to molecule generation in this setting. They are applicable both to models with and without a latent space structure.

2.3 Binding affinity estimation

In drug discovery, compounds are expected to bind to a given target that can be a protein molecule, an RNA molecule or bigger complexes. To this end, it is crucial to study the binding mode of small compounds and to simulate their binding affinity. Binding affinity estimates can be difficult to obtain. We have three avenues for obtaining such data :

- Experimental bio-assays consist in in-vitro quantitative assessment of the interaction between a compound and a target. This does not require computing and is reliable, but often scarce.
- Quantitative Structure Activity Relationship (QSAR) models are machine learning models trained on experimental bio-assays to derive structure-activity rules. They generalize quite well on some targets but as any other machine learning algorithm, they have a validity domain. This means that their accuracy for a compound $\mathbf{x}$ depends on the similarity between their training set and $\mathbf{x}$
- Molecular docking softwares search for ligand conformations that minimize the binding energy with a given protein pocket. Although the estimates of the binding affinity they provide are noisy, the top-scoring compounds are enriched in active molecules [31]. Thus they are widely used to select the most promising leads in a library. Docking is computationally very intensive ($\sim$ 10 CPU minutes / compound). The small proportion of molecules for which the docking gives insightful results, added to the time cost of docking computations,

make it crucial to carefully choose the compounds to dock. The euclidean molecular latent space can be used to design an iterative strategy for this purpose.

2.4 Binding affinity optimization

Since the binding affinity to one or several targets ultimately yields the phenotypical effect of the compound, it is natural to try to generate compounds with high binding affinities. In [27], authors train a GAN to sample from a distribution resembling the one of known actives in latent space, and evaluate the samples affinities using a QSAR model. Alternatively, a QSAR model can be used to bias the generation. This is the approach used in the RL framework by [8, 17]. This method gives good results, but is based on a QSAR model that can be inaccurate: the authors constrain the model to avoid drifting away from the validity region of the model. This is what we are trying to circumvent by directly using a docking program, that uses physics-based molecular mechanics force fields to compute binding affinities. We use docking as an oracle and include docking score in the function $\mathbf{f}$. However this oracle is now costly and the sparse rewards induced by the RL frameworks are not tractable, which calls for specific methods to train generative models. We explore two of them: Bayesian Optimization (BO) that is well established, but has limitations in high dimensions and the recently published Conditioning by Adaptive Sampling (CbAS).

3 Methods

3.1 Graph to sequence Variational Autoencoder

Inspired by previous VAE architectures, we propose a graph to sequence Variational Autoencoder that achieves comparable performance to state of the art models, while benefiting from the following design choices :

- Using the molecular graph as input and a graph convolution encoder solves the issue of data augmentation. It is also better suited for learning a chemically organized latent space, since chemically similar molecules have very similar graphs, while their SMILES representations may change more due to syntax rules. Finally, graph convolution embeddings and circular fingerprints were shown to enhance molecular properties predictions [32, 33].
- On the decoding side, however, decoding to a molecular graph results in more complex architectures than decoding to sequences, thereby increasing the computation time. The main weakness of string-based decoders is that due to the sensitivity of the SMILES syntax, only a small fraction of the decoded sequences resulted in a valid molecule. By using recently published Selfies [3], we circumvent this issue and generate 100% valid molecules.

We train our model on the Moses benchmark set [34], which contains ~ 1.5M molecules from the ZINC database [35] clean leads. The model architecture and training regime are detailed in A.1.

3.2 Dopamine Receptor D3 Activity models

Experimental assays and QSAR activity model The Excape database [36] contains 3482 active molecules for human Dopamine Receptor 3, and more than 300k inactives. QSAR activity models leverage the structure of known actives and inactives to learn a decision function. In [17], the authors build an activity prediction model for human Dopamine Receptor 2, which has strong structural similarity to Dopamine Receptor 3. As the model performs well, we follow the same architecture and training procedure to train a SVM for DRD3 activity prediction. Model training and performances are detailed in Supplementary Material A.2.

Molecular docking We use Autodock Vina [37] to estimate compounds binding affinities to human Dopamine Receptor D3. We use the PDB structure of the receptor provided in the DUD-E dataset [38], and keep the same binding site coordinates. We set the exhaustiveness of the conformation space search to 16, as a reasonable compromise between running time and enrichment factor. We compute the docking score as the average of the 10 best poses found for each ligand.

3.3 Lead Generation

To generate promising leads for a target, requires sampling in the chemical space from a distribution $q(\mathbf{z})$ that maximizes the docking score oracle $\mathbf{f}$, while generating a diversity of molecules. Reinforcement learning methods require many evaluations of $\mathbf{f}$, and therefore seem ill-suited. It is worth noting that $\mathbf{f}$ is not differentiable, which also excludes gradient-based methods.

3.3.1 Bayesian Optimization

Bayesian optimization uses Gaussian processes as a surrogate model for black box functions that are costly to evaluate. It enables to optimize the objective in a query-efficient way, by sampling points that maximize expected improvement. However, it has inherent limitations for the lead generation problem we attempt to solve. To take samples, some kind of rigid (not learnt nor adaptive) sampling is generally used, meaning that the expected improvement under the Gaussian process model is computed over each point of a grid of a certain resolution.

This computation of the expected improvement over the samples space does not scale well to a high number of dimensions for a large batch size as the grid evaluation becomes intractable (This amounts to finding an estimate on all molecules and only picking the most promising candidates for docking). In addition, BO was shown to perform better when the latent space is shaped by the objective function. This is not the case for binding affinities, since the chemical space is likely to exhibit activity cliffs and scattered activity peaks. The final goal is to be able to take samples from the model, and this method uses rejection sampling to generate favorable samples. This choice limits scalability when sampling tens of thousands of compounds. As it is a well-established baseline, we implement it for comparison.

3.3.2 Conditioning by Adaptive Sampling

To address the limitations of Bayesian Optimization, we turn to a recently published method : Conditioning by Adaptive Sampling (CbAS) [30]. This method trains a generative model that also seeks to maximize an objective function. This methods uses a prior generative model and shifts its distribution to maximize an expectation. Queries are used in an efficient way thanks to an importance sampling scheme coupled with reachable objectives for the model. The alternating phases of tuning and sampling also enable a more efficient implementation.

CbAS starts with a prior generative model with parameters $\boldsymbol{\theta}^{(0)}$, $p(\mathbf{x}|\boldsymbol{\theta}^{(0)})$ and the optimization is formulated by conditioning this probability on the random variable $\boldsymbol{S}, p(\mathbf{x}|\boldsymbol{S}, \boldsymbol{\theta}^{(0)})$. This random variable represents the values for a probabilistic (or noisy) oracle : $\boldsymbol{S} = (\mathbf{f}(\mathbf{x}) > \gamma)|\mathbf{x}$. We now see that this conditioned probability model is a distribution that maximizes the function $\mathbf{f}$ when γ goes to the maximum value of the function. However this conditional probability is intractable and the authors propose to use variational inference to approximate it. The parametric family is a generative model with parameters $\boldsymbol{\phi}$, $q(\mathbf{x}|\boldsymbol{\phi})$ that solves :

$$\begin{aligned}\boldsymbol{\phi}^* &= \underset{\boldsymbol{\phi}}{\text{argmin}}\, D_{KL}\left(p(\mathbf{x}|\boldsymbol{S}, \boldsymbol{\theta}^{(0)})||q(\mathbf{x}|\boldsymbol{\phi})\right) \\ &= \underset{\boldsymbol{\phi}}{\text{argmax}}\, \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta}^{(0)})}\left[p(\boldsymbol{S}|\mathbf{x})\text{log}(q(\mathbf{x}|\boldsymbol{\phi})\right]\end{aligned}$$

The authors then use importance sampling instead of always sampling from the prior to estimate this expectation. The proposal distributions are the successive generative models obtained at each iteration. The last key idea is to use a fixed quantile of the successive generative models distributions as a value for $\gamma^{(t)}$, to set reachable objectives for the model : $\boldsymbol{S}^{(t)} = \mathbf{f}(\mathbf{x}) > \gamma^{(t)}$. A detailed derivation can be found in the original paper and result in solving for $\boldsymbol{\phi}^{(t)}$ for each t in the following equation :

$$\boldsymbol{\phi}^{(t+1)} = \underset{\boldsymbol{\phi}}{\text{argmax}}\, \mathbb{E}_{q(\mathbf{x}|\boldsymbol{\phi}^{(t)})}\left[\frac{p(\mathbf{x}|\theta^{(0)})}{q(\mathbf{x}|\boldsymbol{\phi}^{(t)})}p(\boldsymbol{S}^{(t)}|\mathbf{x})\text{log}(q(\mathbf{x}|\boldsymbol{\phi})\right]$$

As a prior model $p(\mathbf{x}|\boldsymbol{\theta}^{(0)})$, one can either use a model trained on a broad chemical space, or leverage previously discovered actives to narrow-down the chemical space by fine tuning the prior on the actives. Then, we take samples from the search model (initialized with the prior), use an oracle with a noise model to get $p(\boldsymbol{S}^{(t)}|\mathbf{x})$ and fine tune the search model with the adequately re-weighted samples.

3.3.3 Efficient implementation

To our knowledge, this is the first use of CbAs with a costly oracle. We already mentioned the computational cost of docking, so using CbAS required a dedicated implementation, alternating between a phase of sampling and re-weighting compounds, a phase of docking and a phase of model training. We have implemented a parallel version of the code that can leverage a multi-node architecture cluster for the docking phase using several hundreds of CPU cores and running the sampling and training phases on GPU nodes. The implementation is available on GitHub. This implementation made the training run in approximately 15 hours using 200 cores. We note that parallelism over more CPU would result in linear speed up, up to several thousands of CPU.

4 Results

4.1 Graph to Selfies VAE

Moses[34] is the reference work for benchmarking molecular generative models. Moses defines a standard train/test split of a given data set and a standard way of computing metrics. Our model achieves state of the art performances while benefiting from design choices in our application. Noteworthy, our model has 3.2M trainable parameters, compared to 5.1M for the default JTVAE implementation [6] and is faster from an algorithmic point of view. We achieve comparable results on the Moses benchmark metrics (Table 1, see Supplementary Table A.1 for full comparison).

Model	Valid	Unique 1k	Unique 10k	IntDiv	IntDiv2	Filters	Novelty
JTN-VAE	1.0	1.0	0.9992	0.8512	0.8453	**0.9778**	**0.9153**
graph2selfies	1.0	1.0	**0.9998**	**0.8560**	0.8496	0.9557	0.9097

Table 1: Moses metrics for samples generated by JTVAE and graph2selfies

4.2 Docking and QSAR on Human Dopamine Receptor 3

We evaluate the ability of molecular docking to enrich a ranked list of molecules by computing enrichment factors at different thresholds. We find that the top-scoring compounds are significantly enriched in actives. The distribution of docking scores for 380 clustered actives [36] and 20 000 decoys is shown in Supplementary Figure A.3. As a control for lead generation experiments, we also dock 10% of the Moses training compounds [34]. Their docking scores distribution is significantly shifted from the actives distribution, and only a negligible fraction of them have a score better than -11 kcal.mol^{-1} (Supplementary Figure A.3).

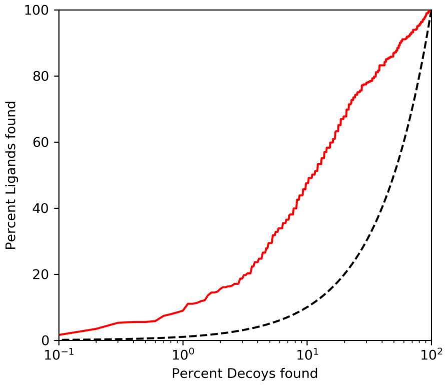

Figure 2: Log ROC curve for ExCAPE database DRD3 actives and inactives

EF	Threshold
13.373	0.1%
11.122	0.5%
8.168	1%
7.229	2%
8.919	1% decoys

Table 2: Enrichment Factor (EF) in active compounds at different thresholds

The QSAR model performs well on the ExCAPE database actives, with a high ROC-AUC. However we argued that these model have a limited validity domain. This is illustrated by comparing the QSAR activiy predictions and docking scores on the Moses training set [34], which contains significantly different compounds from the ExCAPE database actives (Supplementary figure A.2). We only consider the most active compounds according to the docking simulations. At this threshold, there is a significant chance for these compounds to be actives. However we see that a lot of these compounds are labeled as negative by the QSAR model, which illustrate this limited domain capacity. In contrast, docking programs have a broad applicability.

4.3 Lead generation

4.3.1 Bayesian Optimization

Bayesian optimization of Quantitative Drug Likeliness (QED) score results in significantly higher-scoring samples after 50 steps (Figure 3). Details about the Bayesian optimization settings are in Supplementary A.4. However, binding affinity optimization, either using QSAR score or docking score, did not show any results after the same number of steps. This may be due to the fact that the latent space is better organized by QED than by binding affinities. It further motivates the use of a generative model to learn a distribution of affine compounds.

We note that the number of samples taken here is one order of magnitude lower than for CbAS, but the sampling for Bayesian optimization cannot support batches of 1000 samples. Moreover this method would definitely not be suitable for extensive sampling of tens of thousands of compounds once trained.

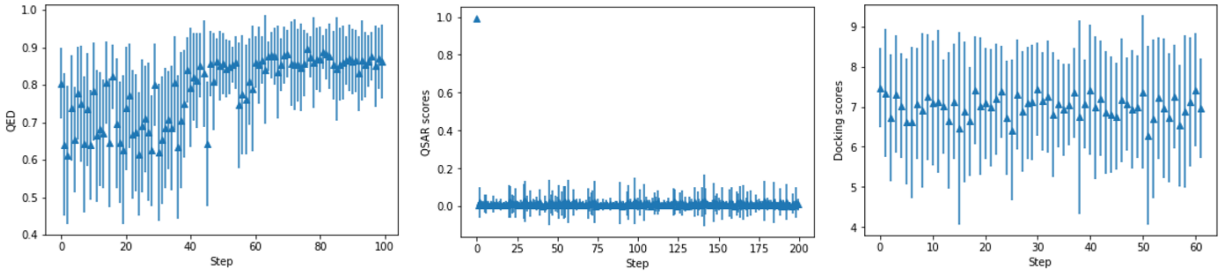

Figure 3: Mean and standard deviation of new samples scores at each step for QED, QSAR score using QSAR activity model and docking scores. In contrast with QED, Bayesian Optimization was not able to find higher scoring samples for QSAR, even when initialized with actives. Likewise, docking scores distributions remain similar along the optimization process

4.3.2 Conditioning by Adaptive Sampling

QED optimization We first apply CbAS to the toy task of enhancing the QED score. This results in a significant shift in the distribution (improving the mean QED from 0.79 to 0.86) while keeping some diversity as is shown in fig A.4. However, the compounds in the Moses data set are already clean: They are taken from the ZINC clean leads and were filtered via medicinal chemistry filters and PAINS filters [39]. This explains the high average QED in the prior. For this reason we can only get a limited improvement. Other papers trying to improve this property are using a prior trained on 250k random compounds from ZINC, thus a much less challenging starting point. For further versions we plan to train another prior on this data set to be able to compare against other methods more easily.

Docking optimization We now come to the main results of our work : The generative model learned using CbAS generates samples with enhanced docking score (shifting the average score from -7.5 to -8.5 kcal.mol^{-1}) while preserving chemical diversity as is shown in fig 4. This means that we managed to generate a population with a docking score distribution close to the actives one, while being able to sample diverse compounds.

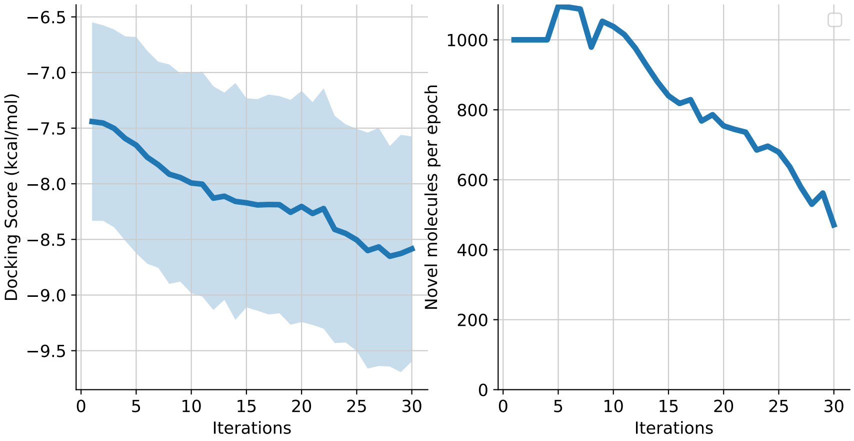

Figure 4: *Left*: The average and standard deviation of the docking scores obtained at each CbAS iteration. *Right*: The number of compounds never sampled before, sampled at each CbAS iteration.

Sampling We want to assess if we are able to generate a population from the new generative model, and if this population has better affinities than the baseline. We easily sampled 100k unique compounds in about one hour after the model was fine tuned. The distribution of docking score of a sub-sample of 4000 molecules from this sampling is available in fig 5. We see that the compounds drawn from the generative model have a significantly better score than random, closer to the active ones (respective means and standard deviations -8.5 +/-1.1, -7.5 +/- 0.8 and -9.6 +/- 1.2 kcal.mol^{-1}).

We also check the internal diversity of the compounds did not collapse during optimization. We use the mean pairwise Tanimoto distance as a diversity metric. This distance is a bit-to-bit distance over fingerprints, ranging from zero to one. Table 3 shows the results we got for random samples taken in Moses compared to ones derived from CbAS sampling. We can see that there is no major difference, indicating the samples produced by our model remain chemically diverse.

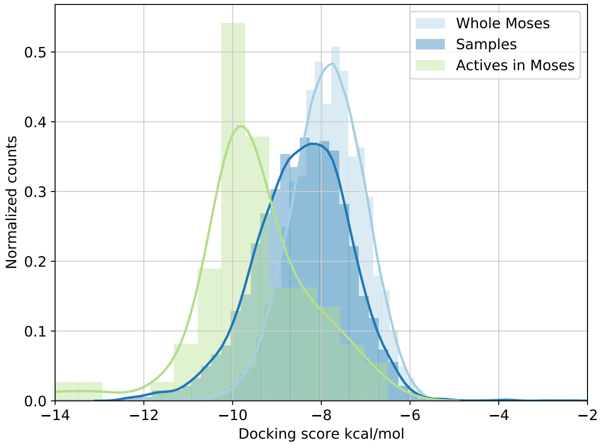

Figure 5: Distributions of scores of docking for three populations : random samples of Moses, samples generated with CbAS and experimental actives in Moses

	Tanimoto distance
Random Sampling	0.89
CbAS Sampling	0.87

Table 3: Mean pairwise Tanimoto distance in 4000 samples produced either by uniform sampling over Moses or by CbAS sampling

We have also assessed whether the samples deteriorated during sampling. Indeed to get unique samples, we used rejection sampling making the first samples the most likely under the generative model. We found that the difference between the first and the last of the 100k samples was statistically significant, but relatively small with respective means of -8.6 and -8.4 kcal.mol^{-1} respectively. The respective distributions are available in Supplementary figure A.5. The 50 top-scoring molecules from 4k random samples drawn from the CbAS generative model are shown in Supplementary Figure A.6. We note the presence of macrocycles among them, which may be actives but are not drug-like molecules. This suggests the use of a composite objective function, to penalize the docking score by drug-likeliness or synthetic accessibility.

These results highlight the potential of our method to speed-up lead discovery without making compromises on the exploration of the chemical space. Indeed, in our experimental setting where Moses is used as the prior chemical space, only 0.62% of the compounds have a docking score better than -10 kcal/mol (estimated on a sample of 10% of Moses). In contrast, at the cost of docking 30k compounds to train the CbAS generative model, we are able to sample from a distribution where 7.4% of the samples have docking scores better than -10. If we were to dock 100k random molecules from the prior chemical space, we therefore would expect to find 620 compounds with a score better than -10. Training the CbAS generative model required 30k docking queries, but the expected number of hits with score better than -10 if we then dock an additional 70k compounds is 5180, almost 10 times higher.

Conclusion

In this paper, we have introduced a new latent space representation for small molecules using graphs and Selfies and show that we obtain state of the art representation with lower computational cost. Using this model, we introduce a new task of building a generative model to create samples with enhanced docking scores, in a feedback loop manner. We show that we are able to train this model so to generate compounds with better and better docking scores. This method could benefit ligand based pipelines by generating compounds with enhanced activity towards a target. It could also benefit structure-based pipelines by generating list of promising compounds for docking, avoiding useless computations on low affinity compounds. Finally it can be seen as a step towards using both approaches in a unified drug discovery pipeline.

We plan to extend the validation by retraining of the prior on other data sets. Smoothness of the objective function may impact the success of the method. Future work could investigate latent space reshaping, using contrastive methods and molecules 3D properties.

Acknowledgments

We thank Jean Philippe Vert, Olivier Sperandio, Arnaud Blondel, Guillaume Bouvier and Laura Ortega for helpful feedback and discussions.

Funding

V.M. is funded by the "Inception Program" and benefits from support from the CRI through "Ecole Doctorale FIRE – Programme Bettencourt". J.B. is funded by McGill Center for Bioinformatics and McGill University School of Computer Science.

References

[1] Jarmila Husby, Giovanni Bottegoni, Irina Kufareva, Ruben Abagyan, and Andrea Cavalli. Structure-based predictions of activity cliffs. *Journal of Chemical Information and Modeling*, 55(5):1062–1076, 2015.

[2] Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noe, and Djork-Arne Clevert. Efficient Multi-Objective Molecular Optimization in a Continuous Latent Space, April 2019.

[3] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry. *CoRR*, abs/1905.13741, 2019.

[4] Jordan Hoffmann, Louis Maestrati, Yoshihide Sawada, Jian Tang, Jean Michel Sellier, and Yoshua Bengio. Data-driven approach to encoding and decoding 3-d crystal structures. *arXiv preprint arXiv:1909.00949*, 2019.

[5] Moritz Hoffmann and Frank Noé. Generating valid euclidean distance matrices. *arXiv preprint arXiv:1910.03131*, 2019.

[6] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2018.

[7] Rim Assouel, Mohamed Ahmed, Marwin H Segler, Amir Saffari, and Yoshua Bengio. Defactor: Differentiable edge factorization-based probabilistic graph generation. *arXiv preprint arXiv:1811.09766*, 2018.

[8] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[10] Rafael Gómez-Bombarelli, David K. Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016.

[11] Robin Winter, Floriane Montanari, Frank Noe, and Djork-Arne Clevert. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.*, 10:1692–1701, 2019.

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

[13] Denis Kuzminykh, Daniil Polykovskiy, Artur Kadurin, Alexander Zhebrak, Ivan Baskov, Sergey Nikolenko, Rim Shayakhmetov, and Alex Zhavoronkov. 3d molecular representations based on the wave transform for convolutional neural networks. *Molecular pharmaceutics*, 15(10):4378–4385, 2018.

[14] Miha Skalic, José Jiménez, Davide Sabbadin, and Gianni De Fabritiis. Shape-based generative modeling for de novo drug design. *Journal of Chemical Information and Modeling*, 59(3):1205–1214, 2019. PMID: 30762364.

[15] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[16] Daniel Schwalbe-Koda and Rafael Gómez-Bombarelli. Generative models for automatic chemical design. *arXiv preprint arXiv:1907.01632*, 2019.

[17] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48, 2017.

[18] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in neural information processing systems*, pages 6410–6421, 2018.

[19] 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.

[20] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder, 2017.

[21] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. *CoRR*, abs/1802.08786, 2018.

[22] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design, 2017.

[23] Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabás Póczos, Jeff Schneider, and Eric P. Xing. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. *ArXiv*, abs/1908.01425, 2019.

[24] Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi S. Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. *CoRR*, abs/1812.01070, 2018.

[25] Farhan Damani, Vishnu Sresht, and Stephen Ra. Black box recursive translations for molecular optimization, 2019.

[26] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures, 2020.

[27] Oleksii Prykhodko, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1):74, 2019.

[28] Anvita Gupta and James Zou. Feedback gan for dna optimizes protein functions. *Nature Machine Intelligence*, 1(2):105–111, 2019.

[29] David H. Brookes and Jennifer Listgarten. Design by adaptive sampling. *CoRR*, abs/1810.03714, 2018.

[30] David H Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. *arXiv preprint arXiv:1901.10060*, 2019.

[31] Niu Huang, Brian K Shoichet, and John J Irwin. Benchmarking sets for molecular docking. *Journal of medicinal chemistry*, 49(23):6789–6801, 2006.

[32] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019.

[33] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[34] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Sergey I. Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *CoRR*, abs/1811.12823, 2018.

[35] Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

[36] Jiangming Sun, Nina Jeliazkova, Vladimir Chupakhin, Jose-Felipe Golib-Dzib, Ola Engkvist, Lars Carlsson, Jörg Wegner, Hugo Ceulemans, Ivan Georgiev, Vedrin Jeliazkov, et al. Excape-db: an integrated large scale dataset facilitating big data analysis in chemogenomics. *Journal of cheminformatics*, 9(1):17, 2017.

[37] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[38] Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012.

[39] Jonathan B. Baell and Georgina A. Holloway. New substructure filters for removal of pan assay interference compounds (pains) from screening libraries and for their exclusion in bioassays. *Journal of Medicinal Chemistry*, 53(7):2719–2740, 2010.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[41] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J Smola, and Zheng Zhang. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[42] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: Programmable bayesian optimization in pytorch, 2019.

A Supplementary Material

A.1 Model architecture and training

The encoder consists in 3 Relational-GCN layers of hidden size 32, with skip connections, resulting in 96-dimensional embeddings. Two dense layers map to the mean and log standard deviation of the latent embeddings, of dimension 56. The decoder is a 3-layer GRU with hidden states of dimension 450. The model was implemented in PyTorch [40] and DGL [41]. The model was trained for 50 epochs using Adam optimizer, a learning rate of 10^{-3} and an exponential decay with rate 0.9 every 40k steps. Batch size was set to 64. For the first 40k steps, we train the model only on reconstruction loss, and then progressively increase the weight of the KL term by 0.02 every 2k steps, until it reaches 0.5

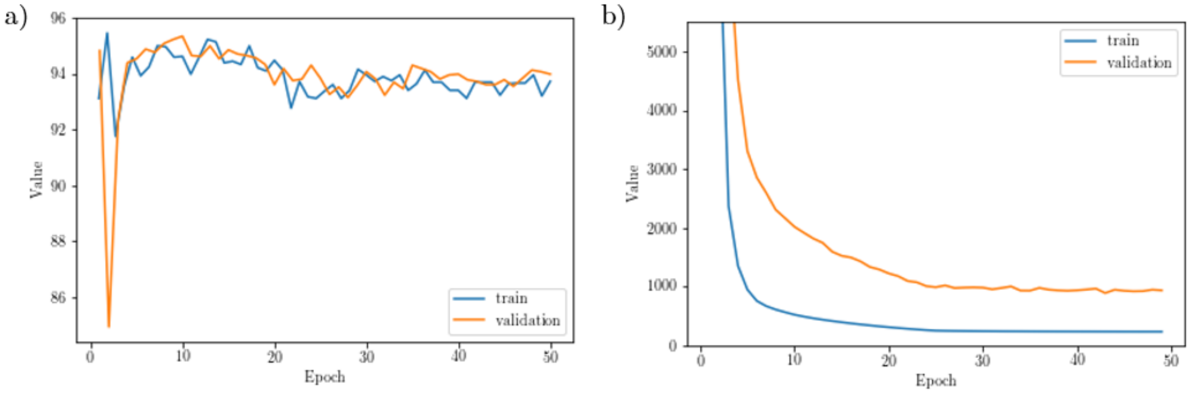

Figure A.1: Fraction of correctly reconstructed characters (a) and KL divergence term (b) during model training, for training and validation set

Table A.1: Moses metrics for all models benchmarked in Moses[34] and graph2selfies

Model	Valid	Unique 1k	Unique 10k	IntDiv	IntDiv2	Filters	Novelty
HMM	0.0760	0.6230	0.5671	0.8466	0.8104	0.9024	**0.9994**
NGram	0.2376	0.9740	0.9217	0.8738	0.8644	0.9582	0.9694
Combinatorial	0.9979	0.9983	0.9948	0.8812	**0.8741**	0.7912	0.9913
CharRNN	0.9748	**1.0000**	**1.0000**	0.8562	0.8503	0.9943	0.8419
AAE	0.9368	**1.0000**	0.9973	0.8557	0.8499	0.9960	0.7931
VAE	0.9767	**1.0000**	0.9984	0.8558	0.8498	**0.9970**	0.6949
JTN-VAE	**1.0000**	**1.0000**	0.9992	0.8512	0.8453	0.9778	0.9153
LatentGAN	0.8970	**1.0000**	0.9970	0.8565	0.8504	0.9727	0.9488
graph2selfies	**1.0000**	**1.0000**	0.9998	0.8560	0.8496	0.9557	0.9097

A.2 QSAR activity model

The Excape database [36] contains 3482 active molecules for human Dopamine Receptor 3, and more than 300k inactives. We follow the procedure in [17] to split the actives into train, validation and test sets: First, we cluster the actives using rdkit's Morgan fingerprints ($r = 3$) and the Butina algorithm with a similarity cutoff of 0.4. We then sort the clusters by size and iteratively assign them to the test, validation and train (4 clusters assigned at a time) sets. The average Tanimoto similarity to the nearest neighbour in the train set, for test molecules, is 0.5465 under this procedure, compared to 0.73 in a random split. We randomly sample 100k inactives and assign them in the same proportions, giving train, validation and test sets with respectively $\frac{4}{6}$, $\frac{1}{6}$ and $\frac{1}{6}$ of the compounds.

We then train a SVM with radial kernel on the training set, and adjust the parameters C and γ by a grid search over the validation set. The model's performance with optimal parameters $C = 32, \gamma = 1e - 2$ is detailed in Table A.2.

Table A.2: DRD3 activity model performance on train, validation and test sets

Model	Train	Valid	Test
ROC-AUC	1.0000	0.9955	0.9942
Precision	0.9996	0.9565	0.9723
Recall	0.9992	0.8551	0.8157

Fig A.2 shows the distribution of the predicted score by the QSAR model, on the compounds with the best score for docking. We see that there is an overlap but that there are lots of compounds that are flagged as negative. Even though they could be false positive for the docking program, it is unlikely that it would be the case for all of them as at these level of score, docking tend to be quite reliable. For this reason we think some of them are false negative for the QSAR model, illustrating the limited domain of validity of this model.

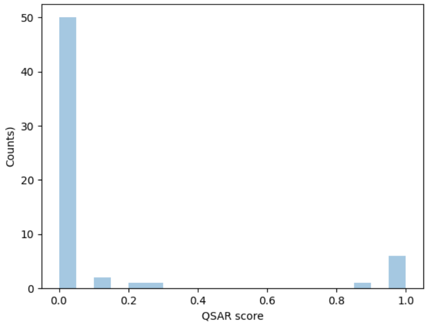

Figure A.2: Distribution of the QSAR scores over the compounds flagged as actives by the docking (energy below 11kcal.mol^{-1})

A.3 Docking scores distributions

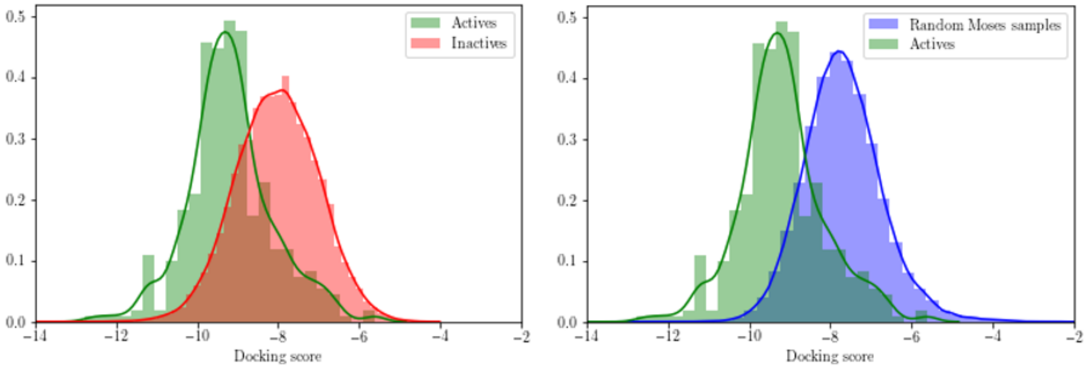

Figure A.3: Distribution of docking scores of active molecules vs ExCAPE inactives *(left)* and active molecules vs random Moses compounds *(right)*

A.4 Bayesian Optimization

Bayesian optimization was implemented using BoTorch [42]. A Gaussian process was trained to predict the objective on 500 initial samples selected to be maximally diverse in the Moses training set. The Gaussian process was then trained for 20 steps by sampling a batch of 50 compounds using Expected Improvement as the acquisition function. For the optimization of QSAR activity scores, the 500 initial samples were selected as maximally diverse from the union of Moses and the QSAR train actives.

A.5 CbAS

Fig A.4 shows the evolution of the CbAS on QED. This is run on Moses data set which explains the high starting value of 0.79. The model is still capable of enriching the compounds a bit, but these high values start to be scarce in the data set and the model cannot simultaneously get higher values while preserving diversity.

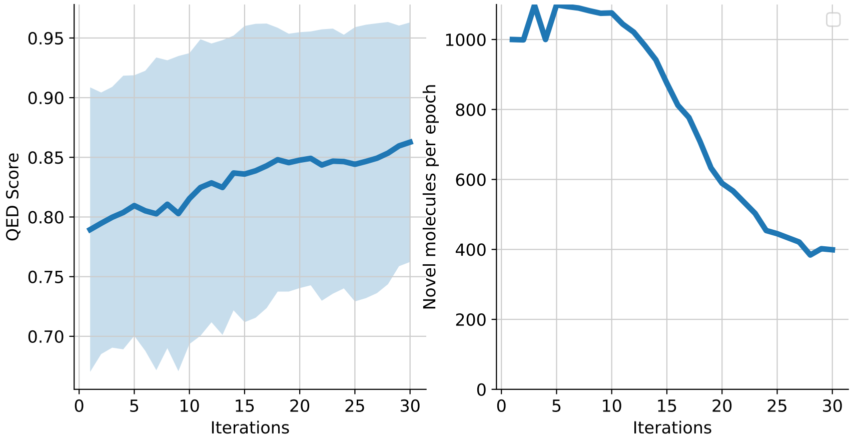

Figure A.4: *Left*: The average and standard deviation of the QED scores obtained at each CbAS iteration. *Right*: The number of compounds never sampled before, sampled at each CbAS iteration.

Model optimization We have tried tuning several parameters and found the training dynamics quite subtle, as too much training increased the scores but crashed the diversity and too little did not optimize the objective functions. We have tuned the number of samples at each epochs, the number of epochs, the usage of teacher forcing, different noise model for the oracle, the number of epochs, the optimizer used (Adam or SGD), the scheduler used, the initial learning rate, the impact of the quantile picked as well as using a non-decreasing γ. We found that that the best combination was reached using 30 epochs of 1000 samples, teacher forcing, gaussian noise of variance of the same magnitude as the variance of the data, Adam optimizer with no scheduler and default learning rate, 6th quantile and a non-decreasing γ.

Fig A.5 shows the distributions of docking scores for the first new samples from the generative model and the last ones. As stated in the main text, the distributions are statistically significant (p-value of 10^{-14}) but small compared to the difference with uniform sampling over Moses.

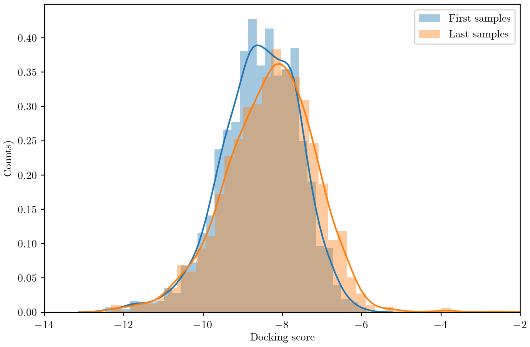

Figure A.5: Distributions of docking scores for the first 2000 samples versus the last 2000 samples

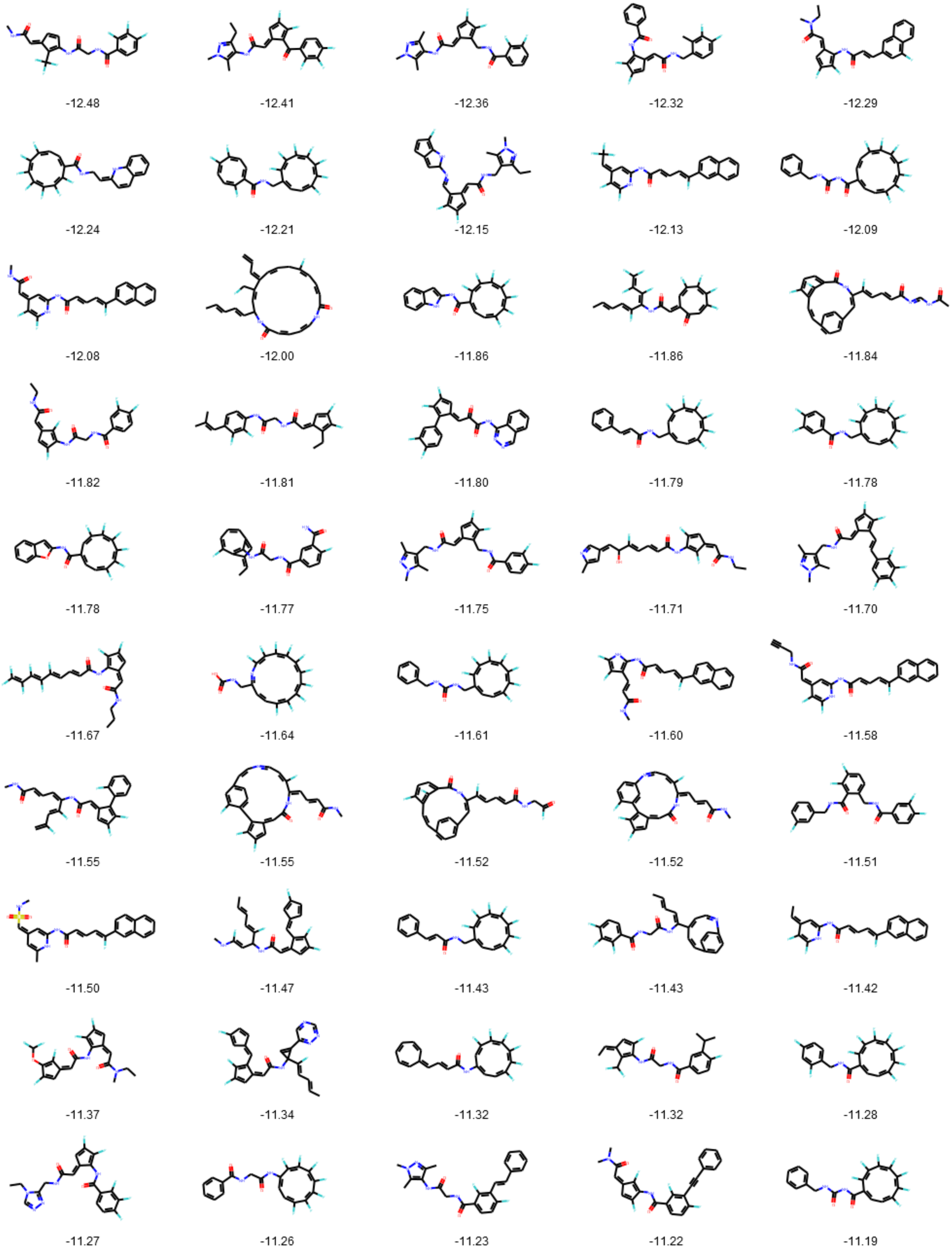

Figure A.6: 50 best scoring molecules in a random sample of 4000 compounds drawn for the CbAS generative model