https://doi.org/10.1186/s13321-024-00887-0

Journal of Cheminformatics

RESEARCH **Open Access**

Evaluation of reinforcement learning in transformer-based molecular design

Jiazhen He[1*], Alessandro Tibo[1], Jon Paul Janet[1], Eva Nittinger[2], Christian Tyrchan[2], Werngard Czechtizky[2] and Ola Engkvist[1,3]

Abstract

Designing compounds with a range of desirable properties is a fundamental challenge in drug discovery. In preclinical early drug discovery, novel compounds are often designed based on an already existing promising starting compound through structural modifications for further property optimization. Recently, transformer-based deep learning models have been explored for the task of molecular optimization by training on pairs of similar molecules. This provides a starting point for generating similar molecules to a given input molecule, but has limited flexibility regarding user-defined property profiles. Here, we evaluate the effect of reinforcement learning on transformer-based molecular generative models. The generative model can be considered as a pre-trained model with knowledge of the chemical space close to an input compound, while reinforcement learning can be viewed as a tuning phase, steering the model towards chemical space with user-specific desirable properties. The evaluation of two distinct tasks—molecular optimization and scaffold discovery—suggest that reinforcement learning could guide the transformer-based generative model towards the generation of more compounds of interest. Additionally, the impact of pre-trained models, learning steps and learning rates are investigated.

Scientific contribution

Our study investigates the effect of reinforcement learning on a transformer-based generative model initially trained for generating molecules similar to starting molecules. The reinforcement learning framework is applied to facilitate multiparameter optimisation of starting molecules. This approach allows for more flexibility for optimizing user-specific property profiles and helps finding more ideas of interest.

Keywords Molecular optimization, Scaffold discovery, Transformer, Generative model, Reinforcement learning, Tanimoto similarity, QSAR

Introduction

The design and optimization of compounds towards potential drug candidates is crucial in drug discovery. The main challenges include the large chemical search space [1] and the requirement of optimization towards multiple properties e.g. physicochemical properties, safety, synthetic feasibility and potency against its target. To accelerate the molecular design and optimization process, various deep neural networks have been explored as molecular generative models, e.g. recurrent neural networks (RNNs) [2–4], variational autoencoders (VAEs) [5–10], transformers [11–14], generative

*Correspondence:
Jiazhen He
jiazhen.he@astrazeneca.com
[1] Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden
[2] Medicinal Chemistry, Research and Early Development, Respiratory and Immunology (R&I), BioPharmaceuticals R&D, AstraZeneca, Gothenburg, Sweden
[3] Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden

adversarial networks (GANs) [15–18], graph neural networks (GNNs) [19–22] and diffusion-based models [23–25]. Early work have been focusing on de novo molecular design which generates molecules from scratch without needing a starting compound, while there is an increasing attention on conditional compound generation and optimization from a specific starting structure that shows promise, e.g. compounds [12, 26–30], scaffolds [14, 22, 31–34] and fragments [25, 35–38]. In this work, we focus on compounds as starting point. In previous publications [12, 29, 30], we treated the molecular optimization problem as machine translation task and trained the transformer model [39] on pairs of similar molecules extracted based on different similarity criteria e.g. Tanimoto similarity on fingerprint, matched molecular pairs and scaffold. The model learns to generate similar molecules to a given input molecule. To generate compounds with desired properties, property change tokens are prepended to the simplified molecular-input line-entry system (SMILES) [40] tokens in order to steer the model towards the chemical space of interest. However, this model is limited to the preselected set of properties during optimization.

Reinforcement learning (RL) [41–43] has been used to guide generative models to explore the chemical space of interest defined by a set of user-defined properties. It provides the flexibility of optimizing molecules towards various user-specified desired properties. Here, we integrate the transformer models [30, 44] trained for generating similar molecules into the REINVENT framework [42] and evaluate the effect of reinforcement learning. Specifically, the evaluation will be conducted on two tasks i.e. molecular optimization and scaffold discovery. Each task will include four example starting molecules with varying level of optimization challenges. The transformer model generates molecules similar to a given starting molecule, and the reinforcement learning is applied to enforce multi-parameter optimisation of the starting molecule. The integration of transformer model, which have learned the surrounding chemical space of input molecules, with RL has potential applicability in the context of constrained optimization of a starting molecule, e.g. molecular optimization and scaffold discovery.

Methods

Transformer based molecular generator

We focus on the transformer models trained on a set of similar molecular pairs. The molecules are represented as SMILES and the SMILES are tokenized to construct a vocabulary, which contains all possible tokens. After training, the models can generate similar molecules to a given input molecule. In particular, two models trained on varying size of training data are examined in REINVENT: the transformer model [30] trained on around 6.5 million molecular pairs extracted from ChEMBL and the transformer model [44] trained on over 200 billion molecular pairs from PubChem. The molecular pairs with a Tanimoto similarity ≥ 0.5 based on RDKit Morgan fingerprints (radius = 2, with counts) are selected. To generate multiple molecules, the non deterministic, multinomial sampling is used. At each time step, a token is randomly selected based on the probability distribution over the vocabulary.

REINVENT

REINVENT [42, 45] is an AI-based tool for molecular design and optimization. It contains three main components: a molecular generative model, a scoring function which scores the generated molecules based on a set of user-specified scoring criteria and produces a combined score as reward, and RL as a search algorithm to steer the generated model towards the chemical space with high reward. Additionally, to reduce the risk of mode collapse and encourage the diversity of the generated molecules, REINVENT uses a molecular memory system called the diversity filter (DF) with different implemented strategies. The DF penalizes the generation of identical compounds or compounds sharing the same scaffold that have been generated too often. The generative model acts as agent and describes the joint probability of generating a molecule represented by a token sequence $T = t_1, t_2, \ldots, t_l$ given an input molecule token sequence X as

$$\mathrm{P}(\mathrm{T}|\mathrm{X};\boldsymbol{\theta}) = \prod_{\mathrm{i}=1}^{\mathrm{l}} \mathrm{P}(\mathrm{t_i}|\mathrm{t_1},\ldots,\mathrm{t_{t-1}},\mathrm{X};\boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{\theta}$ represents the model parameters, t_i represents the i-th token of T, and l represents the length of T. Accordingly, the negative log likelihood (NLL) is defined as

$$\mathrm{NLL}(\mathrm{T}|\mathrm{X};\boldsymbol{\theta}) = -\sum_{\mathrm{i}=1}^{\mathrm{l}} \log \mathrm{P}(\mathrm{t_i}|\mathrm{t_1},\ldots,\mathrm{t_{i-1}},\mathrm{X};\boldsymbol{\theta}) \tag{2}$$

Figure 1 shows the general RL workflow. The agent is initialized using the transformer prior, which generates similar molecules to an input molecule. The reinforcement learning loop is further performed to tune the agent's focus on narrower chemical space of interest defined by a set of user-specified scoring components. Specifically, in each RL step, a batch of molecules (batch size=128) are sampled from the agent given the input molecule, and then evaluated based on the scoring function. The evaluated score is combined with the prior and the agent's negative log

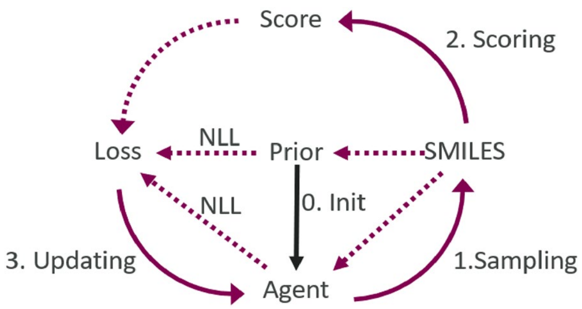


Fig. 1 General RL workflow. The agent is initialized (0) by a transformer prior which learns to generate similar molecules to a given input molecule. The RL loop starts with sampling (1) a batch of molecules represented as SMILES which then are scored based on the set of user-specified scoring components (2). The loss is computed by combining the score and the negative log likelihood of the generated molecules and finally the agent is updated (3) to minimize the loss

likelihood for loss computation. The loss is defined as Eq. 3 following [42].

$$\mathcal{L}(\boldsymbol{\theta}) = \left(\mathrm{NLL_{aug}}(\mathrm{T}|\mathrm{X}) - \mathrm{NLL}(\mathrm{T}|\mathrm{X};\boldsymbol{\theta})\right)^2. \tag{3}$$

$\mathrm{NLL_{aug}}$ represents the augmented negative log likelihood defined as

$$\mathrm{NLL_{aug}}(\mathrm{T}|\mathrm{X}) = \mathrm{NLL}(\mathrm{T}|\mathrm{X};\boldsymbol{\theta}_{\mathrm{prior}}) - \sigma \mathrm{S}(\mathrm{T}) \tag{4}$$

where $S(T) \in [0, 1]$ is a scoring function whose value represents the evaluated desirability of molecule sequence T. It is an aggregation function of multiple scoring components. More details of S can be found in [42]. $\sigma > 0$ is a scalar coefficient balancing the desirability with prior likelihood of a sequence, and $\boldsymbol{\theta}_{\mathrm{prior}}$ are the parameters of the prior. The agent is updated to minimize Eq. 3, as demonstrated previously [32, 42], which encourages increasing the evaluated score while keeping the agent not very far away from the prior which has learnt to produce valid and similar molecules. Note that at the beginning of the training $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{prior}}$, $\boldsymbol{\theta}_{\mathrm{prior}}$ are kept fixed, while $\boldsymbol{\theta}$ are updated.

Experimental setup

The computational experiments aim to evaluate whether RL could improve the performance of transformer-based generative models in generating molecules with desired properties. The evaluation is conducted for two application scenarios,

1 *Scaffold discovery*: generate new scaffold ideas that are active against the dopamine receptor type 2 (DRD2) target.
2 *Molecular optimization*: generate close analogues to improve the activity against the DRD2 target compared to the input molecule.

As a proxy for biologcal activity, we use the DRD2 activity model from Olivecrona et al. [41] which was trained on data extracted from ExCAPE-DB [46]. The output of the model is the predicted probability of a given molecule to be active (pIC50≥5). For both scaffold discovery and molecular optimization, it is common to start with compounds which have already shown reasonable potency. Four compounds were selected from the DRD2 active compounds with pIC50≥5 in ExCAPE-DB as input starting molecules for thorough investigation. Figure 2 shows the input compounds, pIC50, the predicted probabilities to be active P(active) and Quantitative Estimate of Drug-likeness (QED) [47] scores. These compounds were selected to simulate different challenges with respect to input starting structure and property score. Additionally, as a supplementary analysis, 100 compounds were selected from the DRD2 active compounds as input compounds, with each compound having P(active)>0.5 and being randomly chosen from the top 100 most frequent unique scaffolds.

Baseline and REINVENT configuration

The goal is to evaluate whether RL could help steer the transformer-based generative model towards a desirable chemical- and physical-property space. Therefore, the transformer models trained on molecular pairs but without RL serve as baselines. For the main experiments, we use our most recent transformer model [44] which is trained on the PubChem database. For RL, different REINVENT configurations are used, see Table 1.

Scoring components: Since we are interested in generating compounds that are active against the DRD2 target, the DRD2 activity model is added to the scoring function. Additionally, QED is included to prevent the model from generating molecules that have high predicted probability

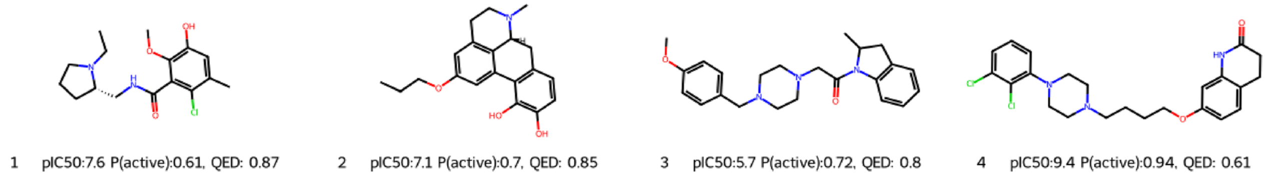

1 pIC50:7.6 P(active):0.61, QED: 0.87
2 pIC50:7.1 P(active):0.7, QED: 0.85
3 pIC50:5.7 P(active):0.72, QED: 0.8
4 pIC50:9.4 P(active):0.94, QED: 0.61

Fig. 2 Input starting molecules. P(active): predicted probability to be active according to the DRD2 activity model

Table 1 Overview of model configuration

Model name	Description			
	RL	Scoring function	DF[1]	Task
No RL (Baseline)	No	NA	NA	Scaffold discovery; Molecular optimization
RL_noDF	Yes	DRD2 activity model; QED	No	Scaffold discovery; Molecular optimization
RL_DF(cmp)	Yes	DRD2 activity model; QED	Compound	Scaffold discovery; Molecular optimization
RL_DF(scaffold)	Yes	DRD2 activity model; QED	Scaffold	Scaffold discovery
RL_DF(cmp)_Sim	Yes	DRD2 activity model; QED Tanimito similarity	Compound	Molecular optimization

[1] DF: Diversity Filter

Table 2 Evaluation metrics

Metrics	Scaffold hopping	Molecular optimization
#Unique compounds with P(active)>0.6 and QED>0.6	✓	
#Unique scaffolds with P(active)>0.6 and QED>0.6	✓	
#Unique generic scaffolds with P(active)>0.6 and QED>0.6	✓	
#Unique compounds with improved P(active) and QED	✓	✓
#Unique scaffolds with improved P(active) and QED	✓	
#Unique generic scaffolds with improved P(active) and QED	✓	
#Unique compounds with improved P(active) and QED; Tanimoto similarity>0.7		✓

to be active but are not drug-like. For the task of molecular optimization, an extra scoring component, Tanimoto similarity based on RDKit Morgan fingerprints (radius = 2, with counts) is added to encourage generating molecules that are similar to corresponding input compound.

Diversity filter: Different diversity filter strategies are used. DF(comp) penalizes the same compound being generated frequently while DF(scaffold) penalizes the compounds sharing the same Murcko type scaffold. For the task of molecular optimization, the option DF(scaffold) is not used since the goal is to generate molecules which are highly similar to the input compound. For comparison, we also include noDF which corresponds to no diversity filter being applied.

The RL loop is run for 1000 steps with each step generating 128 molecules[1], which results in total 128,000 molecules. Therefore, for the baseline model without RL, we sample 128,000 molecules for comparison. Since multinomial sampling is non-deterministic, we run the experiments ten times and report the averaged results with ± one standard deviation for the input starting molecules in Fig. 2.

Evaluation metrics

In general, we are interested in understanding whether RL could help to generate additional, diverse high-scoring compounds. Table 2 shows the evaluation metrics used for the tasks of scaffold discovery and molecular optimization. For scaffold discovery, the focus is to find a novel scaffold which exhibits high chance to be active and good QED score (i.e. P(active)>0.6 and QED>0.6). Additionally, the improved predicted activity and QED over input compounds are examined in a secondary analysis. For molecular optimization, it is favourable to have close analogues to the input molecule with improved predicted probability to be active and improved QED. Here, "scaffold" represents Murcko scaffold from RDKit which removes the side chains and the "generic scaffold" is the Murcko scaffold which converts all atom types to carbon and all bonds to single.

Results and discussion

RL vs No RL for the scaffold discovery task

Most RL settings perform better than No RL in terms of generating molecules with the predicted probability to be active>0.6 and QED>0.6 via all evaluation metrics and for all input compounds (Fig. 3). RL_DF(cmp) generates more unique compounds than RL_noDF, which validates the advantage of penalizing the compounds

[1] The RL process (1000 steps) takes about 30 mins with about 800 MB GPU memory on a single GPU Nvidia Tesla V100.

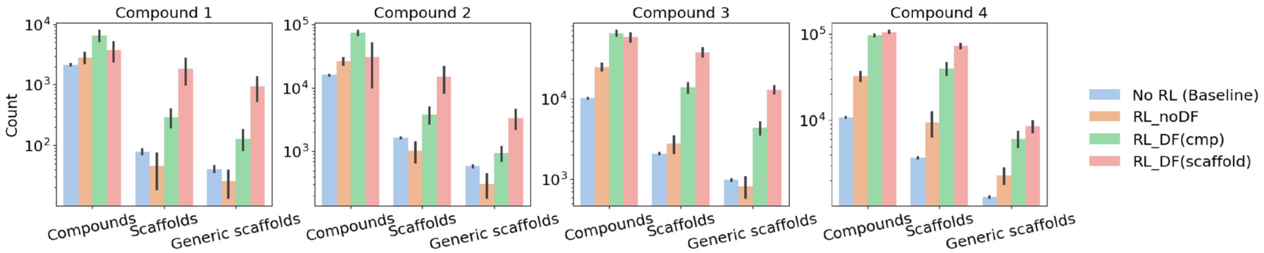


Fig. 3 Scaffold discovery task: mean values and ±1 standard deviation over 10 runs for the number of unique compounds, scaffolds and generic scaffolds that show P(active)>0.6 and QED>0.6. RL generally outperforms No RL, and RL_DF(scaffold) performs best on finding most unique scaffolds and generic scaffolds with desirable properties

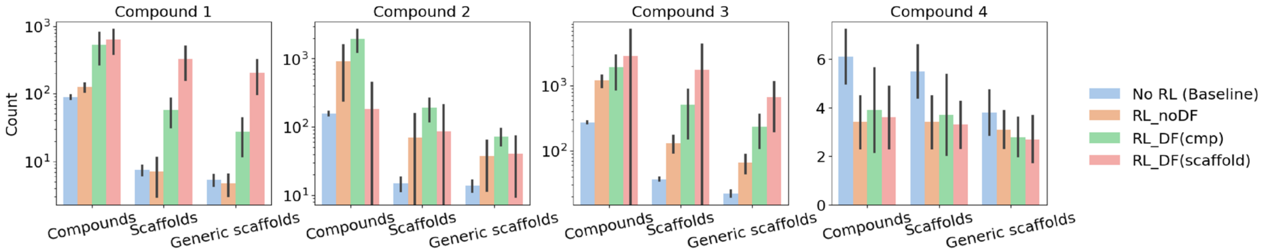


Fig. 4 Scaffold discovery task: mean values and ±1 standard deviation over 10 runs for the number of unique compounds, scaffolds and generic scaffolds that show improved P(active) and QED compared to corresponding input molecule

that have been generated frequently to improve diversity. RL_DF(scaffold) generates more unique scaffolds and generic scaffolds than RL_noDF and RL_DF(cmp), which suggests the benefit of penalizing the frequent generated scaffolds. Especially for scaffold discovery efforts, this can be a useful strategy to increase scaffold diversity.

Furthermore, we examine the performance of achieving higher P(active) and QED upon the input molecules for a secondary analysis. Similar trend can be found that most RL settings perform better than No RL for all input compounds except compound 4 (Fig. 4). The reason why RL struggles with compound 4 might be that the predicted activity for the starting molecule is already very high, which makes it difficult to identify even more potent compounds with RL. Additionally, for compound 2 and compound 4, RL_DF(scaffold) performs worse than RL_noDF and RL_DF(cmp) which indicates changes in the scaffold for these compounds does not improve activity and/or QED. One possible explanation for this is that the scoring function is not set and optimized towards improving predicted activity and QED explicitly. It aims to generate molecules with high scores, but not necessarily higher than the input molecules. This might also contribute to the observed high standard deviation across different runs. Overall, depending on the starting molecules' properties and structural complexity, it is not unexpected to observe different behaviors. For example, for compound 1 with P(active)=0.61, it appears to be easier to improve when exploring diverse scaffolds. While for compound 4 which already has P(active)=0.94, it is difficult to improve and change the scaffold.

RL vs No RL for molecular optimization task

Figure 5 shows the results for the molecular optimization task with molecules achieving higher P(active) and QED scores. Most RL settings perform better than No RL on all evaluation metrics except for compound 4. RL_DF(cmp) generally generates more compounds with improved properties and are less similar to the input molecule, as can be seen from the lower number of compounds with Tanimoto similarity>0.7 in comparison with RL_DF(cmp)_Sim. This indicates that adding Tanimoto similarity to scoring function help generating molecules that are more similar to the input compound, which is useful for local molecular optimization - exploration of the close chemical space of an input compound.

The improvement of RL over No RL is not as large as seen in the scaffold discovery task (Fig. 3). This may be because there is more possibilities in discovering compounds with diverse scaffolds and relatively favorable properties, compared to identifying molecules that closely resemble the input molecule while exhibiting improved properties.

Notably, for compound 4 which has P(active)=0.94, RL (i.e. RL_DF(cmp)_Sim) shows a slight improvement

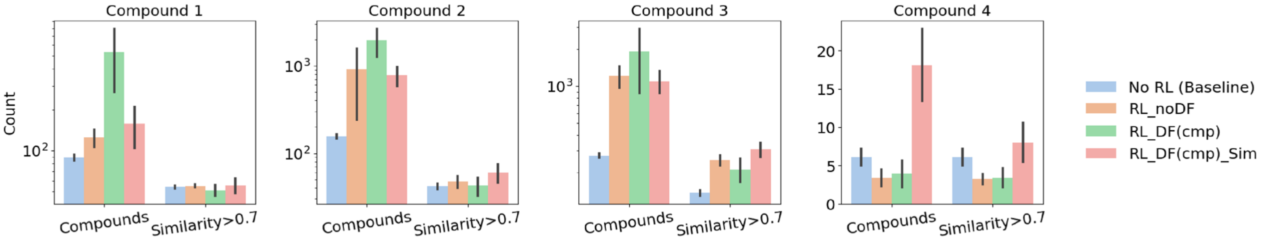


Fig. 5 Molecular optimization task: mean values and ±1 standard deviation over 10 runs for the number of unique compounds that show improved P(active) and QED compared to corresponding input molecule ("Compounds" in Figure) and additionally Tanimoto similarity above 0.7 ("Similarity>0.7" in Figure). RL generally outperforms No RL, and RL_DF(cmp)_Sim performs best in generating compounds with improved properties and Tanimoto similarity above 0.7 compared with corresponding input compound

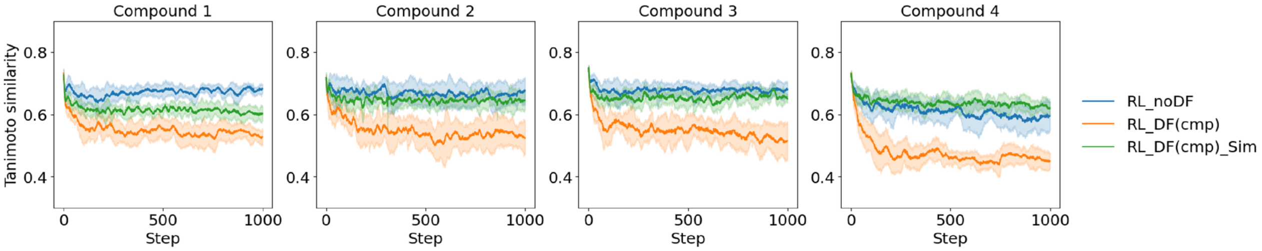


Fig. 6 Molecular optimization task: Tanimoto similarity to input compound per RL step. Results are mean values and ±1 standard deviation over 10 runs

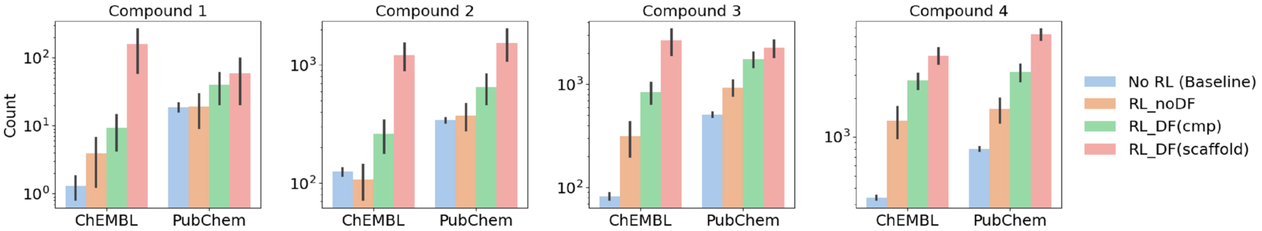


Fig. 7 Scaffold discovery task: effect of pre-trained priors. Results are mean values and ±1 standard deviation over 10 runs for the number of unique scaffolds that show P(active)>0.6 and QED>0.6

over No RL, unlike in the scaffold discovery task (Fig. 4). This might because the Tanimoto similarity scoring component helps the model generate more similar compounds to compound 4, which are also more likely to be highly active.

Figure 6 investigates the Tanimoto similarity to the corresponding input compound changes as RL steps progress. It can be seen that RL_noDF mostly maintains a decent similarity as RL steps increase, while RL_DF(cmp) exhibits a declining similarity but mostly above 0.5. RL_DF(cmp) encourages the generation of more unique compounds with improved properties than RL_noDF (as shown in Fig. 5) at the expense of reduced similarity. RL_DF(cmp)_Sim helps to increase similarity compared to RL_DF(cmp) (Fig. 6) and leads to more unique compounds with similarity above 0.7 and improved properties than RL_noDF and RL_DF(cmp) as shown in Fig. 5.

Effect of pre-trained priors

Here, we evaluate the effect of the priors trained on different sizes of training data, in particular, the transformer model trained on ChEMBL [30] and PubChem [44]. Figure 7 shows the number of unique scaffolds with P(active)>0.6 and QED>0.6 for scaffold discovery task. Without RL, the PubChem prior already yields more compounds of interest than the ChEMBL prior. This could be because the PubChem prior was trained on a much larger scale (200B vs 6.5M). Most RL configurations improve performance for both priors. The PubChem prior consistently outperforms the ChEMBL

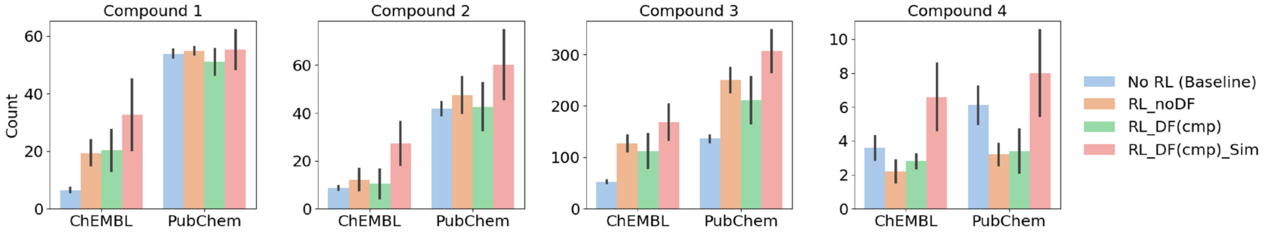


Fig. 8 Molecular optimization task: effect of pre-trained priors. Results are mean values and ±1 standard deviation over 10 runs for the number of unique compounds that show improved P(active) and QED over corresponding input molecule

prior with the exception of RL_DF(scaffold) where ChEMBL prior show comparable performance. This might be because the PubChem prior has a knowledge of closer area of an input molecule than ChEMBL prior, resulting in a slower adaptation of diverse scaffolds generation. Figure 8 shows the results for molecular optimization task. Similarly, the PubChem prior generates more compounds with desirable properties compared to the ChEMBL prior. In general, RL facilitates the generation of more compounds with desirable properties for both priors, with the PubChem prior typically outperforming the ChEMBL prior in the evaluated tasks.

Effect of learning steps

Here, we evaluate the effect of varying number of RL learning steps, i.e. 100, 1000 and 2000 steps. Figure 9 shows the number of unique scaffolds with P(active)>0.6 and QED>0.6 for scaffold discovery task when varying the number of learning steps. For simplicity, we only show RL_DF(scaffold). It can be seen that RL_DF(scaffold) exhibits a consistent trend of generating more unique scaffolds with desirable properties as the number of steps increases, while No RL shows limited improvement. This is expected because without RL, the same area of chemical space is searched every step, whereas RL allows the agent to update at each step, exploring different regions. A similar trend can be found for the molecular optimization task in Fig. 10. With more steps, RL_DF(cmp)_Sim tends to generate more unique compounds with improved properties that are similar to input molecule. Overall, these findings suggest that increasing the number of learning steps typically leads to the discovery of more compounds of interest.

Effect of learning rates

Here, we evaluate the effect of different learning rates, i.e. 0, 1e-5, 1e-4 (default) and le-3. Notebly learning rate=0 is equivalent to No RL. Figure 11 shows the number of unique scaffolds with P(active)>0.6 and QED>0.6 for scaffold discovery task with increasing learning rate. For simplicity, we focus on RL_DF(scaffold). It can be seen that as the learning rate increases up to 1e-4, more scaffolds with desirable properties are found, indicating the model is guided more efficiently towards the desired chemical space. Meanwhile, the variance between different runs increases. This may be because with a higher learning rate, each update to the model parameters is larger, directing the model's focus towards a more different region of the chemical space. A too high learning rate i.e. le-3 in this study, results in noisy and unstable update. Figure 12a–d shows the overlap of three runs for the unique compounds generated by RL_DF(scaffold) for compound 1. It shows a tendency of reduced overlap as the learning rate increases, indicating each run tends to explore different parts of the chemical space. Additionally, a larger chemical space is explored with a higher learning rate of up to 1e-4. These factors might contribute

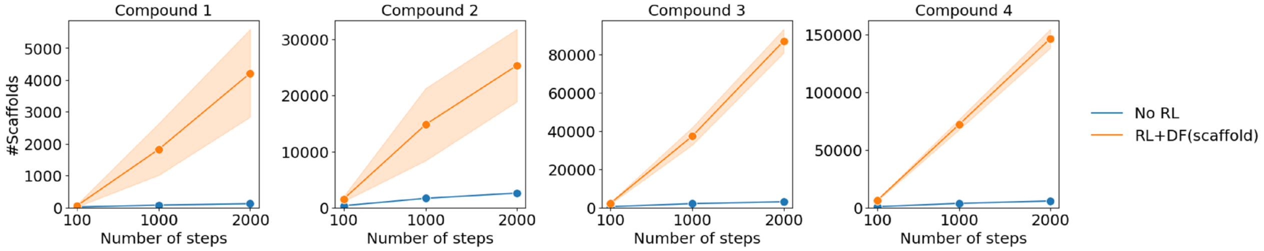


Fig. 9 Scaffold discovery task: effect of learning steps on the number of unique scaffolds with P(active)>0.6 and QED>0.6. Results are mean values and ±1 standard deviation over 10 runs. RL_DF(scaffold) consistently generates more unique scaffolds with desirable properties as the number of steps increases

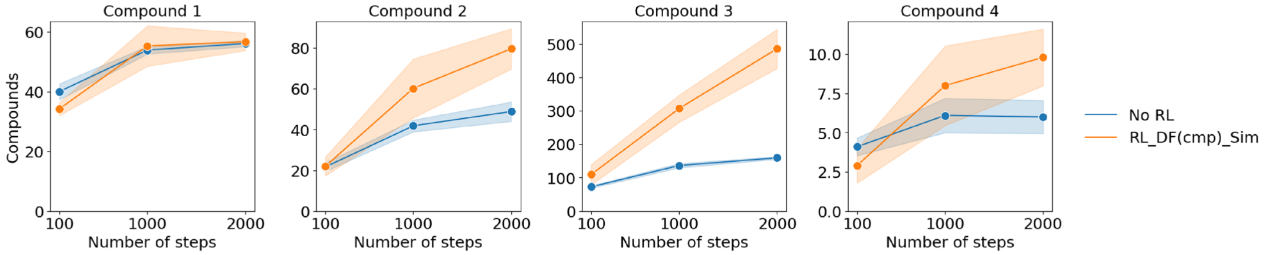


Fig. 10 Molecular optimization task: effect of learning steps on the number of unique compounds with improved P(active) and QED score, and a Tanimoto similarity >0.7 compared to corresponding input molecule. Results are mean values and ±1 standard deviation over 10 runs. RL_DF(cmp)_Sim generally produces more unique compounds with improved properties that are similar to input molecule as the number of steps increases

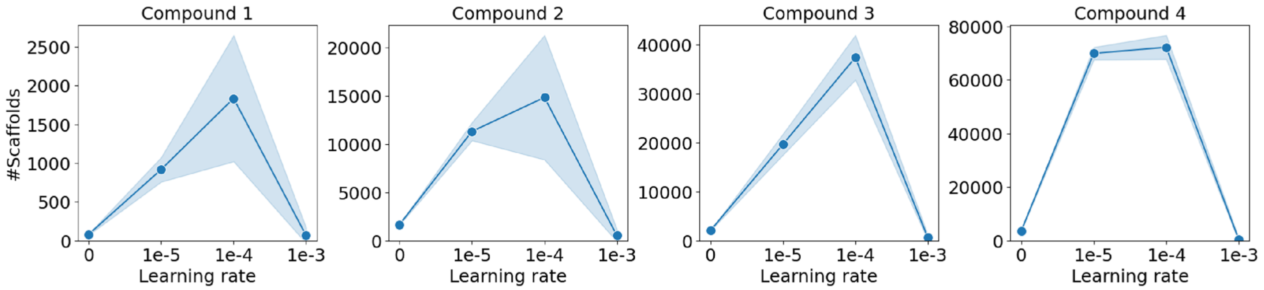
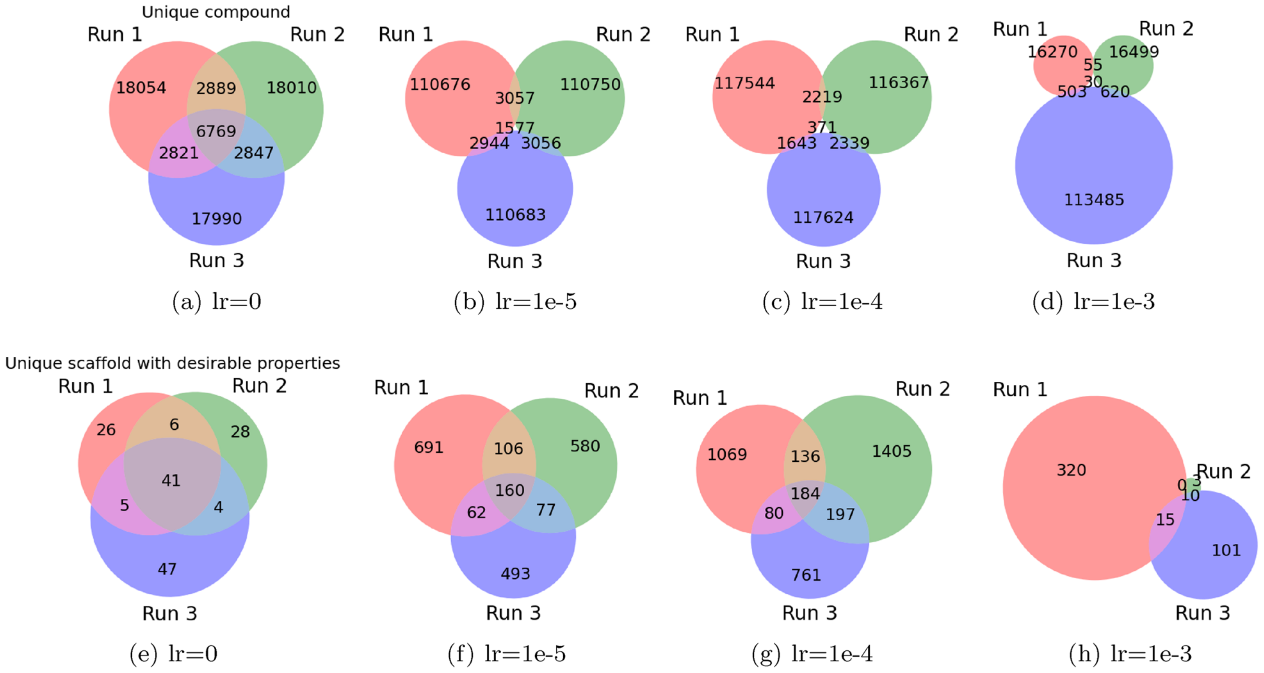


Fig. 11 Scaffold discovery task: effect of learning rates on the number of unique scaffolds with P(active)>0.6 and QED>0.6 when using RL_DF(scaffold). Results are mean values and ±1 standard deviation over 10 runs. Learning rate=0 is equivalent to No RL. Increasing learning rate (up to 1e-4) tends to explore the desired chemical space more efficiently while introducing higher variance between different runs. Learning rate 1e-3 leads to dramatic decrease in model performance



(a) lr=0 (b) lr=1e-5 (c) lr=1e-4 (d) lr=1e-3

(e) lr=0 (f) lr=1e-5 (g) lr=1e-4 (h) lr=1e-3

Fig. 12 Scaffold discovery task: overlap of three runs with varying learning rates (lr) on the unique compounds (a-d) and unique scaffolds with P(active)>0.6 and QED>0.6 (e-h) produced by RL_DF(scaffold) for compound 1. Generally, higher learning rate (up to 1e-4) results in less overlap chemical space between different runs and exploration of larger chemical space

to the greater variance in the number of unique scaffold with desirable properties between different runs (Figs. 11 and 12e–g). The results for the molecular optimization task can be found in Supplementary Figs. S1 and S2 where similar results are found.

A learning rate of 1e-3 is too high and results in very large differences between different runs (Fig. 12d) and much fewer scaffolds of interest are found (Fig. 12) h. Figure 13 compares the percentage of valid molecules generated by RL_DF(scaffold) for compound 1 when learning rate is 1e-4 and 1e-3. It can be seen that learning rate 1e-4 produces stable output and a high percentage of valid molecules between different runs, while vast variance is observed when the learning rate is 1e-3.

Figure 14 examines the prior NLL distribution of unique compounds generated by RL_DF(scaffold) for compound 1 with varying learning rates. With a higher learning rate (up to 1e-4), a larger chemical space, deviating from prior, is explored. This is because RL helps steering the agent towards the chemical space with favourable properties, potentially directing it away from prior. Consequently, the agent has a higher chance (lower NLL) to generate molecules with desirable properties than the prior (Fig. 14 right).

Figures 15 and 16 show example molecules generated for scaffold discovery and molecular optimization task respectively. We can see that these molecules are typically more likely (lower NLL) to be generated by the agent using RL compared to the prior.

Effect of balancing factor σ

Here, we evaluate the effect of σ in Eq. 4 which balances the desirability of a molecule (enforced by the scoring objective) and the likelihood of this molecule generated from the prior. A default value is 120. Figure 17 shows the results for the molecular optimization task. A lower σ generally results in more unique similar molecules (i.e. Tanimoto similarity > 0.7) for all three RL settings as shown in Fig. 17a. This is because a lower

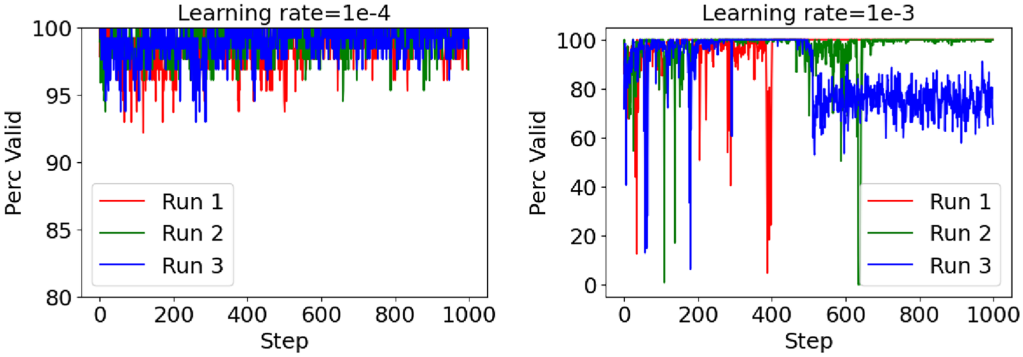


Fig. 13 Percentage of valid molecules generated per RL step produced by RL_DF(scaffold) for compound 1. Too high learning rate 1e-3 results in dramatic instability

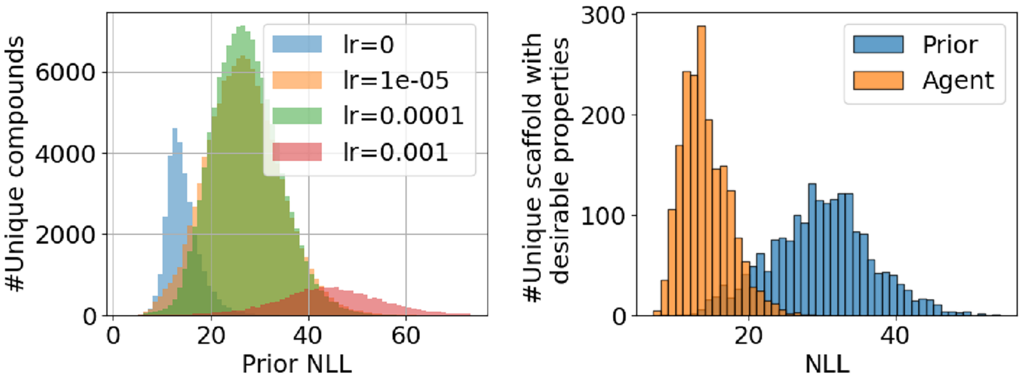


Fig. 14 Left: the prior NLL distribution of unique compounds generated by RL_DF(scaffold) for compound 1 with varying learning rate. Right: the prior and agent NLL distribution of unique scaffold with P(active)>0.6 and QED>0.6 generated by RL_DF(scaffold) for compound 1 when learning rate=1e-4. The lower the NLL of a molecule, the higher the chance to generate

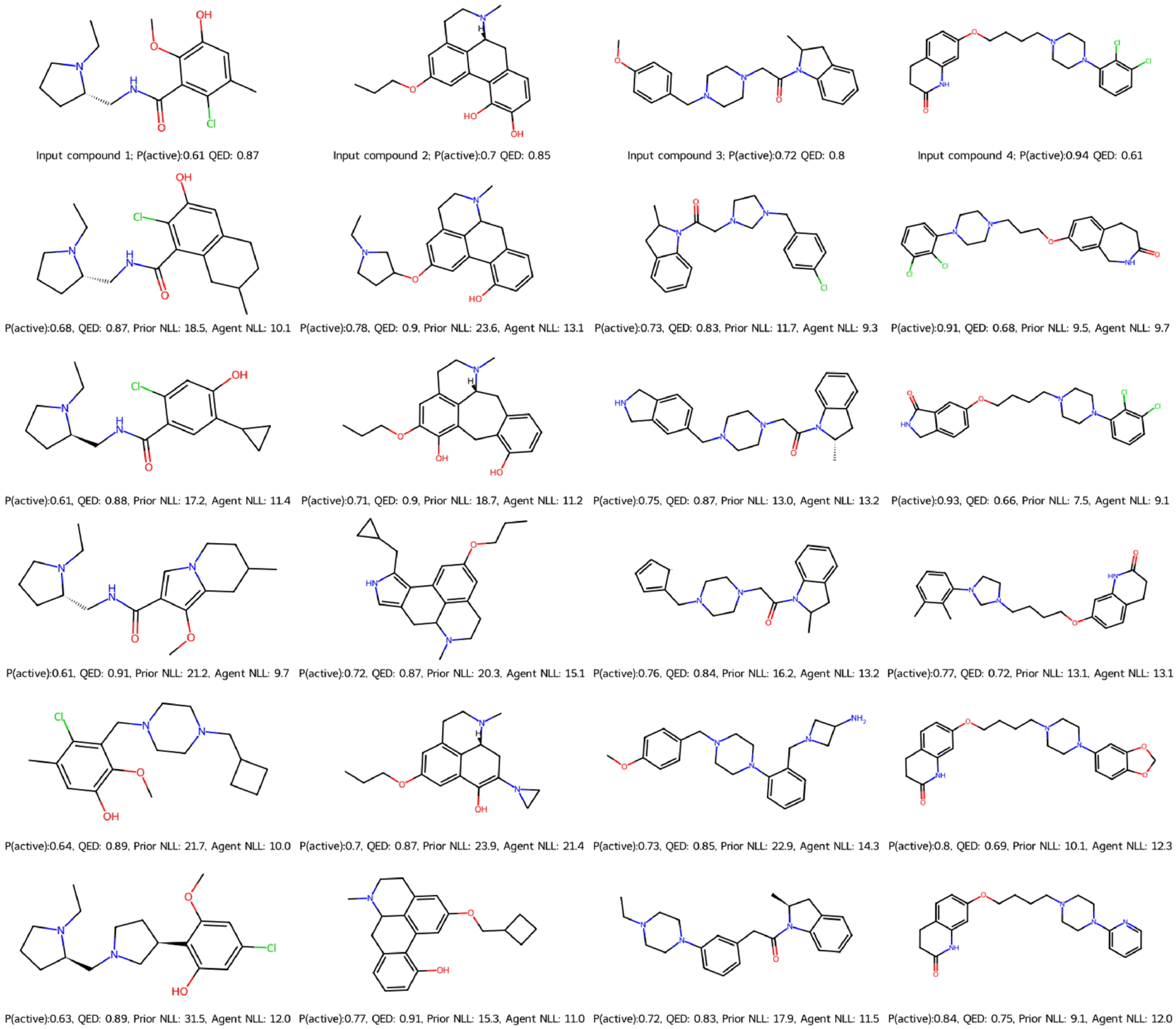

Input compound 1; P(active):0.61 QED: 0.87
Input compound 2; P(active):0.7 QED: 0.85
Input compound 3; P(active):0.72 QED: 0.8
Input compound 4; P(active):0.94 QED: 0.61
P(active):0.68, QED: 0.87, Prior NLL: 18.5, Agent NLL: 10.1
P(active):0.78, QED: 0.9, Prior NLL: 23.6, Agent NLL: 13.1
P(active):0.73, QED: 0.83, Prior NLL: 11.7, Agent NLL: 9.3
P(active):0.91, QED: 0.68, Prior NLL: 9.5, Agent NLL: 9.7
P(active):0.61, QED: 0.88, Prior NLL: 17.2, Agent NLL: 11.4
P(active):0.71, QED: 0.9, Prior NLL: 18.7, Agent NLL: 11.2
P(active):0.75, QED: 0.87, Prior NLL: 13.0, Agent NLL: 13.2
P(active):0.93, QED: 0.66, Prior NLL: 7.5, Agent NLL: 9.1
P(active):0.61, QED: 0.91, Prior NLL: 21.2, Agent NLL: 9.7
P(active):0.72, QED: 0.87, Prior NLL: 20.3, Agent NLL: 15.1
P(active):0.76, QED: 0.84, Prior NLL: 16.2, Agent NLL: 13.2
P(active):0.77, QED: 0.72, Prior NLL: 13.1, Agent NLL: 13.1
P(active):0.64, QED: 0.89, Prior NLL: 21.7, Agent NLL: 10.0
P(active):0.7, QED: 0.87, Prior NLL: 23.9, Agent NLL: 21.4
P(active):0.73, QED: 0.85, Prior NLL: 22.9, Agent NLL: 14.3
P(active):0.8, QED: 0.69, Prior NLL: 10.1, Agent NLL: 12.3
P(active):0.63, QED: 0.89, Prior NLL: 31.5, Agent NLL: 12.0
P(active):0.77, QED: 0.91, Prior NLL: 15.3, Agent NLL: 11.0
P(active):0.72, QED: 0.83, Prior NLL: 17.9, Agent NLL: 11.5
P(active):0.84, QED: 0.75, Prior NLL: 9.1, Agent NLL: 12.0

Fig. 15 Scaffold discovery task: example of generated compounds with P(active)>0.6 and QED>0.6

σ keeps the agent closer to the prior which is trained to generate similar molecules. Meanwhile, a higher σ generally helps generate more unique compounds with improved properties in Fig. 17b. Ultimately, there is no clear trend in the number of unique compounds that are both similar and show improved properties when varying σ in Fig. 17c. Among the three RL settings, RL_DF(cmp)_Sim generates the most unique similar compounds while RL_DF(cmp) generates the most unique compounds with improved properties. This is because without Tanimoto similarity included in the scoring objective, the agent could explore chemical space more freely in search of high scoring compounds, potentially deviating from the prior. In general, RL_DF(cmp)_Sim performs best in finding compounds with both similarity and improved properties. RL_DF(cmp) generates more unique similar compounds than RL_noDF as shown in Fig. 17a indicating the benefit of diversity filter to improve uniqueness. When σ=120, RL_DF(cmp) mostly becomes worse than RL_noDF. This might be because the high σ shifts the agent away from the prior.

Overall, for local molecular optimization, the goal is to generate (1) unique molecules that are (2) highly similar (i.e. Tanimoto similarity > 0.7) to the input molecule while also showing (3) desirable properties (improved properties in this case). Achieving all these criteria is important and challenging since they can be conflicting. The prior, scoring objectives and diversity filter have direct impact on similarity, desirable properties, and

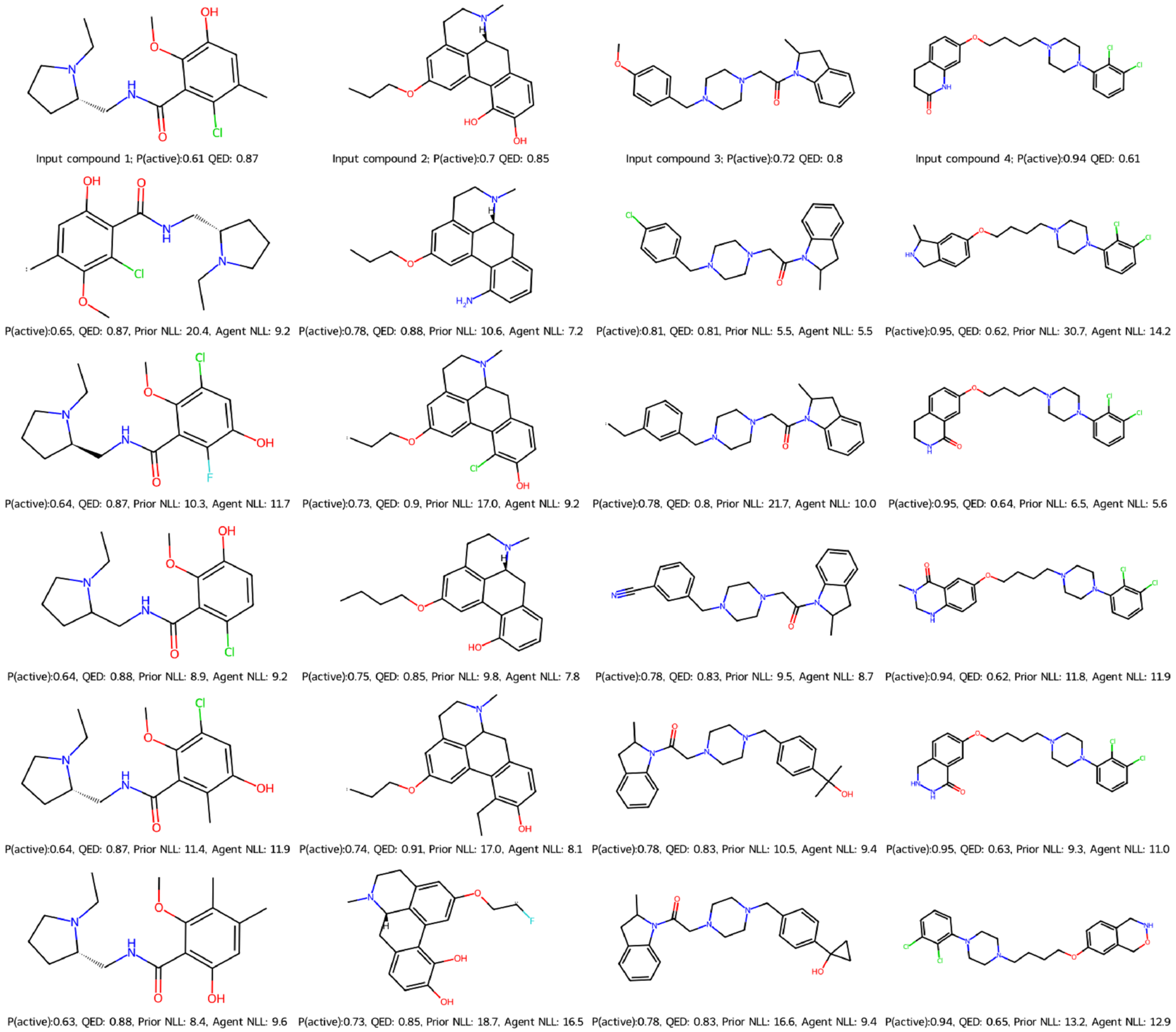

Input compound 1; P(active):0.61 QED: 0.87
Input compound 2; P(active):0.7 QED: 0.85
Input compound 3; P(active):0.72 QED: 0.8
Input compound 4; P(active):0.94 QED: 0.61
P(active):0.65, QED: 0.87, Prior NLL: 20.4, Agent NLL: 9.2
P(active):0.78, QED: 0.88, Prior NLL: 10.6, Agent NLL: 7.2
P(active):0.81, QED: 0.81, Prior NLL: 5.5, Agent NLL: 5.5
P(active):0.95, QED: 0.62, Prior NLL: 30.7, Agent NLL: 14.2
P(active):0.64, QED: 0.87, Prior NLL: 10.3, Agent NLL: 11.7
P(active):0.73, QED: 0.9, Prior NLL: 17.0, Agent NLL: 9.2
P(active):0.78, QED: 0.8, Prior NLL: 21.7, Agent NLL: 10.0
P(active):0.95, QED: 0.64, Prior NLL: 6.5, Agent NLL: 5.6
P(active):0.64, QED: 0.88, Prior NLL: 8.9, Agent NLL: 9.2
P(active):0.75, QED: 0.85, Prior NLL: 9.8, Agent NLL: 7.8
P(active):0.78, QED: 0.83, Prior NLL: 9.5, Agent NLL: 8.7
P(active):0.94, QED: 0.62, Prior NLL: 11.8, Agent NLL: 11.9
P(active):0.64, QED: 0.87, Prior NLL: 11.4, Agent NLL: 11.9
P(active):0.74, QED: 0.91, Prior NLL: 17.0, Agent NLL: 8.1
P(active):0.78, QED: 0.83, Prior NLL: 10.5, Agent NLL: 9.4
P(active):0.95, QED: 0.63, Prior NLL: 9.3, Agent NLL: 11.0
P(active):0.63, QED: 0.88, Prior NLL: 8.4, Agent NLL: 9.6
P(active):0.73, QED: 0.85, Prior NLL: 18.7, Agent NLL: 16.5
P(active):0.78, QED: 0.83, Prior NLL: 16.6, Agent NLL: 9.4
P(active):0.94, QED: 0.65, Prior NLL: 13.2, Agent NLL: 12.9

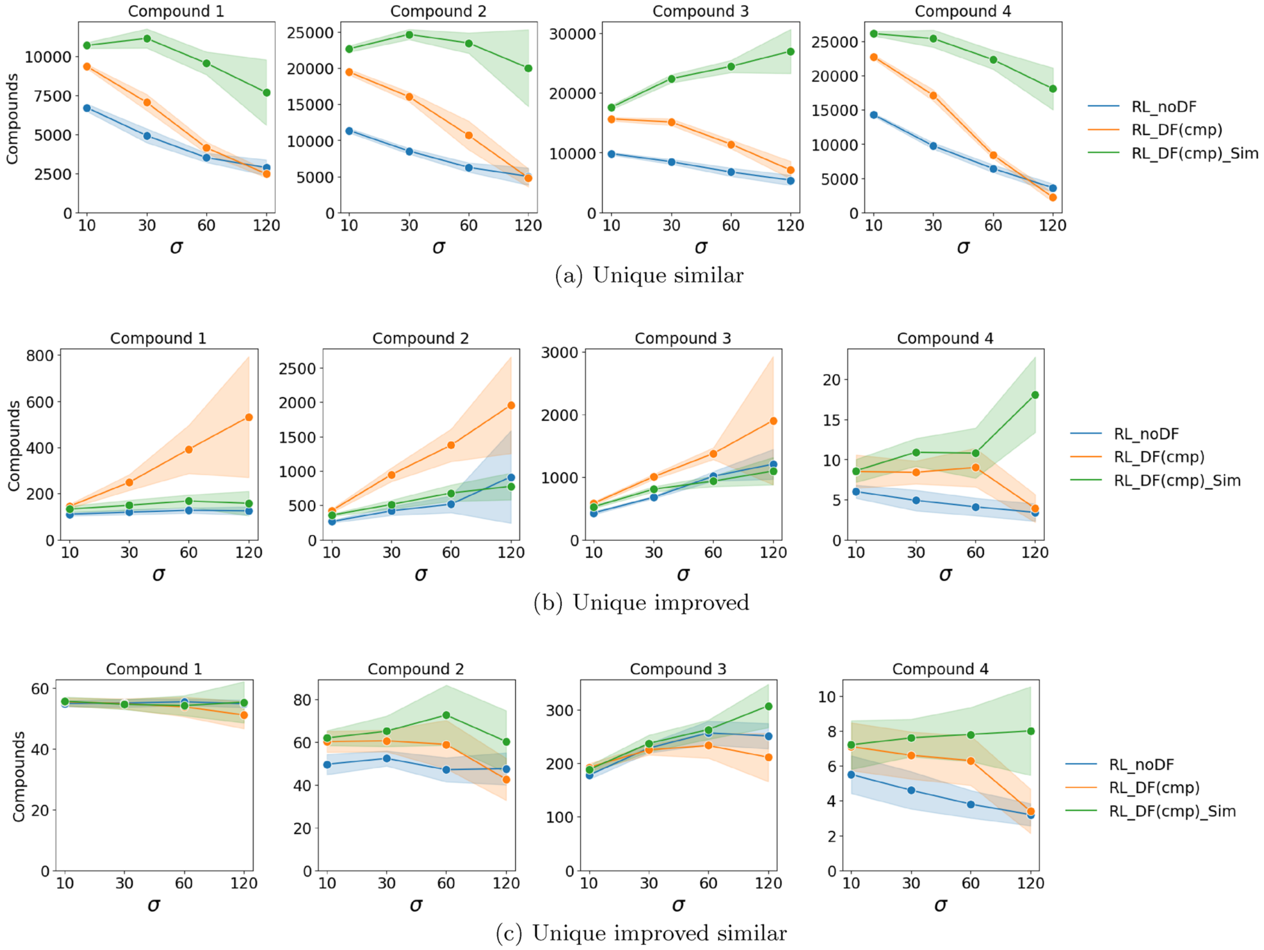
Fig. 16 Molecular optimization task: example of generated compounds with improved P(active) and QED, and Tanimoto similarity >0.7 compared with corresponding input compound

uniqueness respectively. Lowering σ brings the agent closer to the prior, thus producing more similar molecules but also finding less compounds with improved properties. Scoring objectives guide the agent towards the chemical space of high scoring compounds but this could also lead to deviations from the prior. The diversity filter helps in exploring more unique compounds but could also have less similar compounds when $\sigma(=120)$ is high. Therefore, it is crucial to understand and consider the impact of these factors.

Supplementary comparison of RL and No RL

Here, we examine the effect of RL in a larger scale, specifically with 100 input starting molecules. A single run is conducted for each configuration. Figure 18 shows the results for the scaffold discovery task with each point representing the performance for each input starting molecule. Clearly, all three RL settings generate more unique compounds, scaffolds and generic scaffolds that show P(active)>0.6 and QED>0.6 than No RL, with RL_DF(scaffold) performing the best followed by RL_DF(cmp) and RL_noDF. This re validates the advantage of RL over No RL, and the use of diversity filtering which penalizes frequently generated compounds or scaffolds to improve diversity.

Figure 19 shows the results for the molecular optimization task. All three RL settings generate more unique compounds that show improved P(active) and QED



(a) Unique similar

(b) Unique improved

(c) Unique improved similar

Fig. 17 Molecular optimization task: effect of σ on the number of (**a**) unique compounds that have Tanimoto similarity above 0.7 relative to corresponding input compound, (**b**) unique compounds that show improved P(active) and QED compared to corresponding input compound and (**c**) unique compounds that show both improved P(active) and QED and Tanimoto similarity above 0.7. Results are mean values and ±1 standard deviation over 10 runs

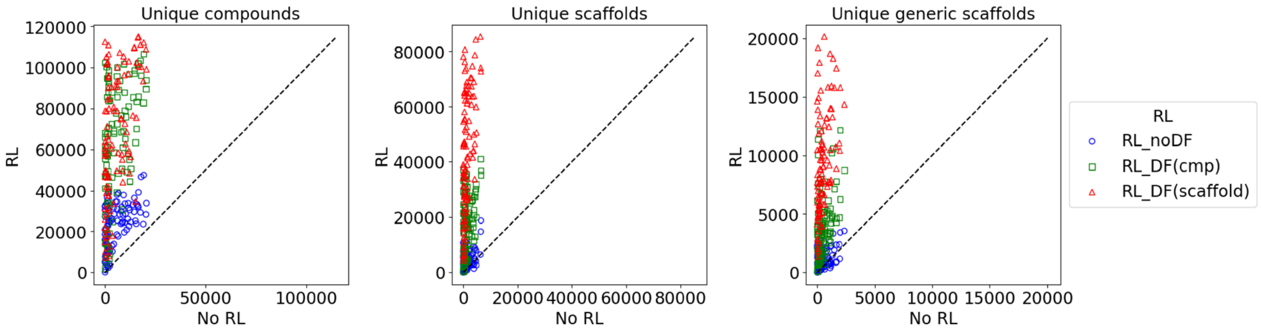


Fig. 18 Scaffold discovery task: comparison of No RL and RL with 100 input starting molecules in terms of generating unique compounds, scaffolds and generic scaffolds that show P(active)>0.6 and QED>0.6. Each point represents the performance for each input starting molecule

than No RL. However, for generating unique similar (*i.e.* Tanimito similarity > 0.7) molecules, RL_noDF and RL_DF(cmp) perform worse than No RL. RL_DF(cmp) generates the most unique compounds with improved properties but least in the unique similar compounds. The reason could be that the agent was guided to focus

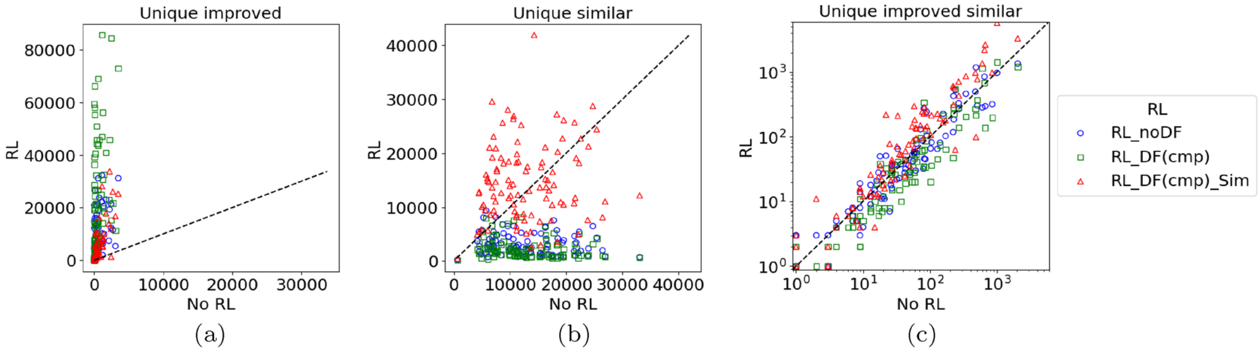


Fig. 19 Molecular optimization task: comparison of No RL and RL with 100 input starting molecules in terms of generating (**a**) unique compounds that show improved P(active) and QED compared to corresponding input compound, (**b**) unique compounds that have Tanimoto similarity above 0.7 relative to corresponding input compound and (**c**) unique compounds that show both improved P(active) and QED and Tanimoto similarity above 0.7. Each point represents the performance for each input starting molecule

on improving properties (enforced by scoring objectives) and unique molecules (enforced by diversity filter) which might deviate from the prior for generating similar molecules. RL_DF(cmp)_Sim which includes Tanimoto similarity as an additional scoring objective help generate more similar compounds as shown in Fig. 19b, and move towards the end goal of both improved properties and similarity in Fig. 19c.

Conclusions

We have evaluated the effect of RL on the transformer-based molecular generative model trained for generating similar molecules to a given input molecule. The generative model serves as a pre-trained model with knowledge of the chemical space surrounding the input molecule, and reinforcement learning acts as fine tuning phase to focus the model on the desirable chemical space based on a set of user-specified property objectives. This provides the flexibility of optimizing molecules towards task-specific property profiles. The evaluation has been performed on two application scenarios, scaffold discovery and molecular optimization. Additionally, the effect of pre-trained priors, learning steps, learning rates and the balancing factor σ was examined. The results have shown that

(i) RL generally helps generate more molecules with desired properties compared to No RL for both scaffold discovery and molecular optimization tasks. Additionally, different behaviors can be expected depending on the starting input molecule's structure and properties, e.g. it can be challenging for RL to find molecules with improved activity if the starting molecule is already highly active.

(ii) RL consistently helps generating more compounds with desirable properties across priors trained on both ChEMBL and PubChem datasets, and the PubChem prior generally outperforms the ChEMBL prior.

(iii) Increasing the number of learning steps typically results in the discovery of more compounds of interest.

(iv) Increasing the learning rate (to a certain extent) tends to explore a larger chemical space and sample the chemical space of interest more efficiently, at the same time a higher learning rate leads to a higher variance between different runs. A too high learning rate can have a dramatic negative impact on the performance.

(v) For the molecular optimization task, a lower σ typically results in more unique similar molecules, whereas a higher σ tends to produce more unique compounds with improved properties. Ultimately, there is no clear trend in the number of unique compounds that are both similar and show improved properties when varying σ.

As an example of optimizing towards user-specified desired properties, we have evaluated how well we can find more active compounds against DRD2 compared to a given starting molecule. However, any property can be optimized in the RL framework as long as it can be used as a scoring function. Notably, the accuracy and generalizability of a predictive model plays an important role in practice.

Our evaluation has been conducted on the tasks of scaffold discovery and molecular optimization. However, it is not limited to these tasks and can be used for molecular generation tasks such as scaffold decorating or fragment linking by adding substructure matching scoring components.

Abbreviations

RNNs	Recurrent neural networks
VAEs	Variational autoencoders
GANs	Generative adversarial networks
GNNs	Graph neural networks
SMILES	Simplified Molecular-Input Line-Entry System
RL	Reinforcement learning
DF	Diversity filter
NLL	Negative log likelihood
DRD2	Dopamine receptor type 2
QED	Quantitative estimate of drug-likeness

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-024-00887-0.

Supplementary Material 1.

Author contributions

J.H. performed the research and wrote the manuscript with help from co-authors. All authors were involved in discussions on the project and revised the manuscript. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The code and models are available at https://github.com/MolecularAI/transformer_rl

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 15 March 2024 Accepted: 21 July 2024
Published online: 08 August 2024

References

1. Polishchuk PG, Madzhidov TI, Varnek A (2013) Estimation of the size of drug-like chemical space based on gdb-17 data. J Comput-Aid Mol Des 27(8):675–679
2. Segler MH, Kogej T, Tyrchan C, Waller MP (2018) Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Central Sci 4(1):120–131
3. Gupta A, Müller AT, Huisman BJ, Fuchs JA, Schneider P, Schneider G (2018) Generative recurrent networks for de novo drug design. Mol Inform 37(1–2):1700111
4. Bjerrum EJ, Threlfall R (2017) Molecular generation with recurrent neural networks (RNNs). arXiv preprint arXiv:1705.04612
5. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, Aguilera-Iparraguirre J, Hirzel TD, Adams RP, Aspuru-Guzik A (2018) Automatic chemical design using a data-driven continuous representation of molecules. ACS Central Sci 4(2):268–276
6. Dai H, Tian Y, Dai B, Skiena S, Song L (2018) Syntax-directed variational autoencoder for molecule generation. In: Proceedings of the International Conference on Learning Representations
7. Lim J, Ryu S, Kim JW, Kim WY (2018) Molecular generative model based on conditional variational autoencoder for de novo molecular design. J Cheminform 10(1):1–9
8. Jin W, Barzilay R, Jaakkola T (2018) Junction tree variational autoencoder for molecular graph generation. In: International Conference on Machine Learning, pp. 2323–2332
9. Liu Q, Allamanis M, Brockschmidt M, Gaunt A (2018) Constrained graph variational autoencoders for molecule design. In: Advances in Neural Information Processing Systems, pp. 7795–7804
10. Simonovsky M, Komodakis N (2018) Graphvae: Towards generation of small graphs using variational autoencoders. In: International Conference on Artificial Neural Networks, pp. 412–422. Springer
11. Bagal V, Aggarwal R, Vinod P, Priyakumar UD (2021) Molgpt: molecular generation using a transformer-decoder model. J Chem Inf Model 62(9):2064–2076
12. He J, You H, Sandström E, Nittinger E, Bjerrum EJ, Tyrchan C, Czechtizky W, Engkvist O (2021) Molecular optimization by capturing chemistâ€™s intuition using deep neural networks. J Cheminform 13(1):1–17
13. Irwin R, Dimitriadis S, He J, Bjerrum EJ (2022) Chemformer: a pre-trained transformer for computational chemistry. Mach Learn Sci Technol 3(1):015022
14. Zheng S, Lei Z, Ai H, Chen H, Deng D, Yang Y (2021) Deep scaffold hopping with multimodal transformer neural networks. J Cheminform 13:1–15
15. Guimaraes GL, Sanchez-Lengeling B, Outeiral C, Farias PLC, Aspuru-Guzik A (2017) Objective-reinforced generative adversarial networks (organ) for sequence generation models. arXiv preprint arXiv:1705.10843
16. Putin E, Asadulaev A, Ivanenkov Y, Aladinskiy V, Sanchez-Lengeling B, Aspuru-Guzik A, Zhavoronkov A (2018) Reinforced adversarial neural computer for de novo molecular design. J Chem Inf Model 58(6):1194–1204
17. Putin E, Asadulaev A, Vanhaelen Q, Ivanenkov Y, Aladinskaya AV, Aliper A, Zhavoronkov A (2018) Adversarial threshold neural computer for molecular de novo design. Mol Pharm 15(10):4386–4397
18. De Cao N, Kipf T (2018) MolGAN: An implicit generative model for small molecular graphs. In: ICML 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models
19. Mercado R, Rastemo T, Lindelöf E, Klambauer G, Engkvist O, Chen H, Bjerrum EJ (2021) Graph networks for molecular design. Mach Learn Sci Technol 2(2):025023
20. Bongini P, Bianchini M, Scarselli F (2021) Molecular generative graph neural networks for drug discovery. Neurocomputing 450:242–252
21. Mahmood O, Mansimov E, Bonneau R, Cho K (2021) Masked graph modeling for molecule generation. Nat Commun 12(1):3156
22. Hu C, Li S, Yang C, Chen J, Xiong Y, Fan G, Liu H, Hong L (2023) Scaffoldgvae: scaffold generation and hopping of drug molecules via a variational autoencoder based on multi-view graph neural networks. J Cheminform 15(1):91
23. Xu M, Yu L, Song Y, Shi C, Ermon S, Tang J (2022) Geodiff: A geometric diffusion model for molecular conformation generation. International Conference on Learning Representations
24. Hoogeboom E, Satorras VG, Vignac C, Welling M (2022) Equivariant diffusion for molecule generation in 3d. In: International Conference on Machine Learning, pp. 8867–8887. PMLR
25. Igashov I, Stärk H, Vignac C, Schneuing A, Satorras VG, Frossard P, Welling M, Bronstein M, Correia B (2024) Equivariant 3d-conditional diffusion model for molecular linker design. Nat Mach Intell 1–11
26. Jin W, Yang K, Barzilay R, Jaakkola T (2018) Learning multimodal graph-to-graph translation for molecule optimization. In: International Conference on Learning Representations
27. Jin W, Barzilay R, Jaakkola T (2019) Hierarchical graph-to-graph translation for molecules. arXiv:1907.11223
28. Jin W, Barzilay R, Jaakkola T (2020) Hierarchical generation of molecular graphs using structural motifs. In: International Conference on Machine Learning, pp. 4839–4848. PMLR
29. He J, Mattsson F, Forsberg M, Bjerrum EJ, Engkvist O, Tyrchan C, Czechtizky W (2021) Transformer neural network for structure constrained molecular optimization. In: ICLR 2021 Workshop: Machine Learning for Preventing and Combating Pandemics
30. He J, Nittinger E, Tyrchan C, Czechtizky W, Patronov A, Bjerrum EJ, Engkvist O (2022) Transformer-based molecular optimization beyond matched molecular pairs. J Cheminform 14(1):18
31. Arús-Pous J, Patronov A, Bjerrum EJ, Tyrchan C, Reymond J-L, Chen H, Engkvist O (2020) Smiles-based deep generative scaffold decorator for de-novo drug design. J Cheminform 12(1):1–18

32. Fialková V, Zhao J, Papadopoulos K, Engkvist O, Bjerrum EJ, Kogej T, Patronov A (2021) Libinvent: reaction-based generative scaffold decoration for in silico library design. J Chem Inf Model 62(9):2046–2063
33. Li Y, Hu J, Wang Y, Zhou J, Zhang L, Liu Z (2019) Deepscaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning. J Chem Inf Model 60(1):77–91
34. Lim J, Hwang S-Y, Moon S, Kim S, Kim WY (2020) Scaffold-based molecular design with a graph generative model. Chem Sci 11(4):1153–1164
35. Guo J, Knuth F, Margreitter C, Janet JP, Papadopoulos K, Engkvist O, Patronov A (2023) Link-invent: generative linker design with reinforcement learning. Digit Discov 2(2):392–408
36. Yang Y, Zheng S, Su S, Zhao C, Xu J, Chen H (2020) Syntalinker: automatic fragment linking with deep conditional transformer neural networks. Chem Sci 11(31):8312–8322
37. Imrie F, Bradley AR, van der Schaar M, Deane CM (2020) Deep generative models for 3d linker design. J Chem Inf Model 60(4):1983–1995
38. Liu X, Ye K, van Vlijmen HW, IJzerman, AP, van Westen, GJ (2023) Drugex v3: scaffold-constrained drug design with graph transformer-based reinforcement learning. J Cheminform 15(1):24
39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008
40. Weininger D (1988) Smiles, a chemical language and information system 1 introduction to methodology and encoding rules. J Chem Inf Comput Sci 28(1):31–36
41. Olivecrona M, Blaschke T, Engkvist O, Chen H (2017) Molecular de-novo design through deep reinforcement learning. J Cheminform 9(1):48
42. Blaschke T, Arús-Pous J, Chen H, Margreitter C, Tyrchan C, Engkvist O, Papadopoulos K, Patronov A (2020) Reinvent 2.0: an ai tool for de novo drug design. J Chem Inf Model 60(12):5918–5922
43. Ghugare R, Miret S (2023) Searching for high-value molecules using reinforcement learning and transformers. arXiv:2310.02902
44. Tibo A, He J, Janet JP, Nittinger E, Engkvist O (2023) Exhaustive local chemical space exploration using a transformer model
45. Loeffler H, He J, Tibo A, Janet JP, Voronov A, Mervin L, Engkvist O (2023) Reinvent4: Modern ai–driven generative molecule design
46. Sun J, Jeliazkova N, Chupakhin V, Golib-Dzib J-F, Engkvist O, Carlsson L, Wegner J, Ceulemans H, Georgiev I, Jeliazkov V (2017) Excape-db: an integrated large scale dataset facilitating big data analysis in chemogenomics. J Cheminform 9:1–9
47. Bickerton GR, Paolini GV, Besnard J, Muresan S, Hopkins AL (2012) Quantifying the chemical beauty of drugs. Nat Chem 4(2):90–98

Publisher's Note