







RESEARCH ARTICLE

Open Access



# Mol-CycleGAN: a generative model for molecular optimization

Łukasz Maziarka<sup>1,2\*</sup> , Agnieszka Pocha<sup>2</sup> , Jan Kaczmarczyk<sup>1</sup> , Krzysztof Rataj<sup>1</sup> , Tomasz Danel<sup>1,2</sup>   
and Michał Warchol<sup>1</sup> 

## Abstract

Designing a molecule with desired properties is one of the biggest challenges in drug development, as it requires optimization of chemical compound structures with respect to many complex properties. To improve the compound design process, we introduce Mol-CycleGAN—a CycleGAN-based model that generates optimized compounds with high structural similarity to the original ones. Namely, given a molecule our model generates a structurally similar one with an optimized value of the considered property. We evaluate the performance of the model on selected optimization objectives related to structural properties (presence of halogen groups, number of aromatic rings) and to a physicochemical property (penalized logP). In the task of optimization of penalized logP of drug-like molecules our model significantly outperforms previous results.

**Keywords:** Drug design, Molecular optimization, Generative models, Deep learning

## Introduction

The principal goal of the drug design process is to find new chemical compounds that are able to modulate the activity of a given target (typically a protein) in a desired way [1]. However, finding such molecules in the high-dimensional chemical space of all molecules without any prior knowledge is nearly impossible. In silico methods have been introduced to leverage the existing chemical, pharmacological and biological knowledge, thus forming a new branch of science—computer-aided drug design (CADD) [2, 3]. Computer methods are nowadays applied at every stage of drug design pipelines [2]—from the search of new, potentially active compounds [4], through optimization of their activity and physicochemical profile [5] and simulating their scheme of interaction with the target protein [6], to assisting in planning the synthesis and evaluation of its difficulty [7].

The recent advancements in deep learning have encouraged its application in CADD [8]. The two main approaches are: *virtual screening*, which uses discriminative models to screen commercial databases and classify molecules as likely active or inactive; *de novo design*, that uses generative models to propose novel molecules that are likely to possess the desired properties. The former application already proved to give outstanding results [9–12]. The latter use case is rapidly emerging, e.g. long short-term memory (LSTM) network architectures have been applied with some success [13–16].

In the center of our interest are the hit-to-lead and lead optimization phases of the compound design process. Their goals are to optimize the drug-like molecules identified in the previous steps in terms of the desired activity profile (increased potency towards given target protein and provision of inactivity towards off-target proteins) and the physicochemical and pharmacokinetic properties. Optimizing a molecule with respect to multiple properties simultaneously remains a challenge [5]. Nevertheless, some successful approaches to compound generation and optimization have been proposed.

\*Correspondence: lukasz.maziarka@ardigen.com

<sup>1</sup> Ardigen, Podole 76, 30-394 Cracow, Poland

Full list of author information is available at the end of the article



© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

In the domain of molecule generation, Recurrent Neural Networks (RNN) still play a central role. They were successfully applied to SMILES, which is a commonly used text representation of molecules [17, 18]. RNN architectures, especially those based on LSTM or GRU, obtain excellent results in natural language processing tasks where the input is a sequence of tokens that varies in length. Unfortunately, generative models built on SMILES can generate invalid sequences that do not correspond to any molecule. Attempting to solve this problem, grammar-based methods were proposed to ensure the correct context-free grammar of the output sequence [18–20]. Another issue with the SMILES representation is its sensitivity to the structure of the represented molecule. Even small changes in the structural formula of a compound can lead to a very different canonical SMILES, which impacts the ordering of atom processing performed by RNNs. Arús-Pous et al. [21] show that randomization of SMILES can substantially improve the quality of generated molecules. Also, several approaches with reinforcement learning at their cores have been used in chemical property optimization [18, 22]. Moreover, RNNs were also successfully applied to molecular graphs, which are in this case constructed node by node [23]. A promising alternative to reinforcement learning is conditional generation, where molecules are generated with the desired properties presented at the input [24, 25].

Variational Autoencoder (VAE) [26] in conjunction with SMILES representation has been used to generate novel molecules from the trained continuous latent space [27, 28]. Additionally, VAE models were also successfully realized directly on molecular graphs [29, 30]. Because of the intermediate continuous representation of the latent space, molecules with similar properties appear in the vicinity of one another. Bayesian optimization can be used to explore this space and find the desired properties [30]. Still, decoding from the latent space is often-times non-trivial and requires to determine the ordering of generated atoms when RNNs are used in this process.

Generative Adversarial Networks (GAN) [31] is an alternative architecture that has been applied to *de novo* drug design. GANs, together with Reinforcement Learning (RL), were recently proposed as models that generate molecules with desired properties while promoting diversity. These models use representations based on SMILES [32, 33], graph adjacency and annotation matrices [34] or are based on graph convolutional policy networks [35]. There are also hybrid approaches which utilize both GANs and latent vector representation in the process of compound generation [36].

To address the problem of generating compounds difficult to synthesize, we introduce Mol-CycleGAN—a generative model based on CycleGAN [37]—extending the scope

of the early version of our method [38] with more advanced experiments and detailed explanations. Given a starting molecule, it generates a structurally similar one but with a desired characteristic. The similarity between these molecules is important for two reasons. First, it leads to an easier synthesis of generated molecules, and second, such optimization of the selected property is less likely to spoil the previously optimized ones, which is important in the context of multiparameter optimization. We show that our model generates molecules that possess desired properties (note that by a molecular property we also mean binding affinity towards a target protein) while retaining their structural similarity to the starting compound. Moreover, thanks to employing graph-based representation instead of SMILES, our algorithm always returns valid compounds.

We evaluate the model's ability to perform structural transformations and molecular optimization. The former indicates that the model is able to do simple structural modifications such as a change in the presence of halogen groups or number of aromatic rings, and we also consider bioisostere replacement, which is relevant to modern drug optimization process. In the latter, we aim to maximize penalized logP to assess the model's usefulness for compound design. Penalized logP is chosen because it is a property often selected as a testing ground for molecule optimization models [30, 35], due to its relevance in the drug design process. In the optimization of penalized logP for drug-like molecules, our model significantly outperforms previous results. Eventually, experiments on increasing bioactivity are conducted with DRD2 as the biological target. To the best of our knowledge, Mol-CycleGAN is the first approach to molecule generation that uses the CycleGAN architecture.

## Methods

### Junction Tree Variational Autoencoder

JT-VAE [30] (Junction Tree Variational Autoencoder) is a method based on VAE, which works on graph structures of compounds, in contrast to previous methods which utilize the SMILES representation of molecules [19, 20, 27]. The VAE models used for molecule generation share the encoder-decoder architecture. The encoder is a neural network used to calculate a continuous, high-dimensional representation of a molecule in the so-called latent space, whereas the decoder is another neural network used to decode a molecule from coordinates in the latent space. In VAEs the entire encoding-decoding process is stochastic (has a random component). In JT-VAE both the encoding and decoding algorithms use two components for representing the molecule: a junction-tree scaffold of molecular sub-components (called clusters) and a molecular graph [30]. JT-VAE shows superior properties compared to SMILES-based VAEs, such as 100% validity of generated molecules.

### Mol-CycleGAN

Mol-CycleGAN is a novel method of performing compound optimization by learning from the *sets* of molecules with and without the desired molecular property (denoted by the sets  $X$  and  $Y$ ). Our approach is to train a model to perform the transformation  $G : X \rightarrow Y$  and then use this model to perform optimization of molecules. In the context of compound design  $X$  and  $Y$  can be, e.g., the set of inactive (active) molecules.

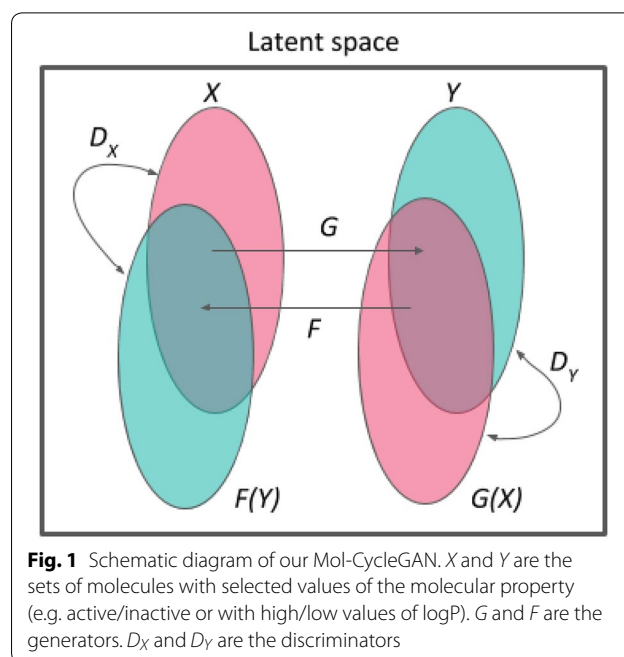
To represent the sets  $X$  and  $Y$ , our approach requires an embedding of molecules which is reversible, i.e. enables both encoding and decoding of molecules.

For this purpose we use the latent space of JT-VAE, which is a representation created by the neural network during the training process. This approach has the advantage that the distance between molecules (required to calculate the loss function) can be defined directly in the latent space. Moreover, molecular properties are easier to express on graphs rather than using linear SMILES representation [39]. One could try formulating the CycleGAN model on the SMILES representation directly, but this would raise the problem of defining a differentiable intermolecular distance, as the standard manners of measuring similarity between molecules (Tanimoto similarity) are non-differentiable.

Our approach extends the CycleGAN framework [37] to molecular embeddings of the latent space of JT-VAE [30]. We represent each molecule as a point in the latent space, given by the mean of the variational encoding distribution [26]. Our model works as follows (Fig. 1): (i) we start by defining the sets  $X$  and  $Y$  (e.g., inactive/active molecules); (ii) we introduce mapping functions  $G : X \rightarrow Y$  and  $F : Y \rightarrow X$ ; (iii) we introduce discriminator  $D_X$  (and  $D_Y$ ) which forces the generator  $F$  (and  $G$ ) to generate samples from a distribution close to the distribution of  $X$  (or  $Y$ ). The components  $F$ ,  $G$ ,  $D_X$ , and  $D_Y$  are modeled by neural networks (see *Workflow* for technical details). The main idea of our approach to molecule optimization is to: (i) take the prior molecule  $x$  without a specified feature (e.g. specified number of aromatic rings, water solubility, activity) from set  $X$ , and compute its latent space embedding; (ii) use the generative neural network  $G$  to obtain the embedding of molecule  $G(x)$ , that has this feature (as if the  $G(x)$  molecule came from set  $Y$ ) but is also similar to the original molecule  $x$ ; (iii) decode the latent space coordinates given by  $G(x)$  to obtain the optimized molecule. Thereby, the method is applicable in *lead optimization* processes, as the generated compound  $G(x)$  remains structurally similar to the input molecule.

To train the Mol-CycleGAN we use the following loss function:

$$L(G, F, D_X, D_Y) = L_{\text{GAN}}(G, D_Y, X, Y) + L_{\text{GAN}}(F, D_X, Y, X) + \lambda_1 L_{\text{cyc}}(G, F) + \lambda_2 L_{\text{identity}}(G, F), \quad (1)$$



**Fig. 1** Schematic diagram of our Mol-CycleGAN.  $X$  and  $Y$  are the sets of molecules with selected values of the molecular property (e.g. active/inactive or with high/low values of  $\log P$ ).  $G$  and  $F$  are the generators.  $D_X$  and  $D_Y$  are the discriminators

and aim to solve

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} L(G, F, D_X, D_Y). \quad (2)$$

We use the adversarial loss introduced in LS-GAN [40]:

$$L_{\text{GAN}}(G, D_Y, X, Y) = \frac{1}{2} \mathbb{E}_{y \sim p_{\text{data}}^Y} [(D_Y(y) - 1)^2] + \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}^X} [(D_Y(G(x)))^2], \quad (3)$$

which ensures that the generator  $G$  (and  $F$ ) generates samples from a distribution close to the distribution of  $Y$  (or  $X$ ), denoted by  $p_{\text{data}}^Y$  ( $p_{\text{data}}^X$ ).

The cycle consistency loss

$$L_{\text{cyc}}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}^Y} [\|G(F(y)) - y\|_1] + \mathbb{E}_{x \sim p_{\text{data}}^X} [\|F(G(x)) - x\|_1], \quad (4)$$

reduces the space of possible mapping functions, such that for a molecule  $x$  from set  $X$ , the GAN cycle brings it back to a molecule similar to  $x$ , i.e.  $F(G(x))$  is close to  $x$  (and analogously  $G(F(y))$  is close to  $y$ ). The inclusion of the cyclic component acts as a regularization and may also help in the regime of low data, as the model can learn from both directions of the transformation. This component makes the resulting model more robust (cf. e.g. the comparison [41] of CycleGAN vs non-cyclic IcGAN [42]). Finally, to ensure that the generated (optimized) molecule is close to the starting one we use the identity mapping loss [37]

$$L_{\text{identity}}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}^Y} [\|F(y) - y\|_1] + \mathbb{E}_{x \sim p_{\text{data}}^X} [\|G(x) - x\|_1], \quad (5)$$

which further reduces the space of possible mapping functions and prevents the model from generating molecules that lay far away from the starting molecule in the latent space of JT-VAE.

In all our experiments we use the hyperparameters  $\lambda_1 = 0.3$  and  $\lambda_2 = 0.1$ , which were chosen by checking a couple of combinations (for structural tasks) and verifying that our optimization process: (i) improves the studied property and (ii) generates molecules similar to the starting ones. We have not performed a grid search for optimal values of  $\lambda_1$  and  $\lambda_2$ , and hence there could be space for improvement. Note that these parameters control the balance between *improvement* in the optimized property and *similarity* between the generated and the starting molecule. We show in the *Results* section that both the improvement and the similarity can be obtained with the proposed model.

**Algorithm 1** Mol-CycleGAN training algorithm. In all experiments in the paper the following values are used:  $\alpha = 0.0001$ ,  $\lambda_1 = 0.3$ ,  $\lambda_2 = 0.1$ ,  $m = 64$ .

**Require:**  $\alpha$  - the learning rate.  $\lambda_1$  - the cycle consistency weight.  $\lambda_2$  - the identity mapping weight.  $m$  - the batch size.

**Require:**  $w_X, w_Y$  - initial discriminators' parameters.  $\theta_G, \theta_F$  - initial generators' parameters.

1: Encode the  $X_{\text{train}}, Y_{\text{train}}$  with the JT-VAE encoder to obtain the latent space representation of molecules from the training dataset.

2: **while**  $\theta_G, \theta_F$  have not converged **do**

3: Sample  $\{x^{(i)}\}_{i=1}^m \sim p_{\text{data}}^X$  a batch from the dataset  $X_{\text{train}}$ .

4: Sample  $\{y^{(i)}\}_{i=1}^m \sim p_{\text{data}}^Y$  a batch from the dataset  $Y_{\text{train}}$ .

5: Calculate the adversarial loss.  $\triangleright$  Eq.(3)

$$L_{\text{GAN}} \leftarrow \frac{1}{2m} \sum_{i=1}^m (D_Y(y^{(i)}) - 1)^2 + \frac{1}{2m} \sum_{i=1}^m (D_Y(G(x^{(i)})))^2 + \frac{1}{2m} \sum_{i=1}^m (D_X(x^{(i)}) - 1)^2 + \frac{1}{2m} \sum_{i=1}^m (D_X(F(y^{(i)})))^2.$$

6: Calculate the cycle consistency loss.  $\triangleright$  Eq.(4)

$$L_{\text{cyc}} \leftarrow \frac{1}{m} \sum_{i=1}^m |F(G(x^{(i)})) - x^{(i)}| + \frac{1}{m} \sum_{i=1}^m |G(F(y^{(i)})) - y^{(i)}|.$$

7: Calculate the identity mapping loss.  $\triangleright$  Eq.(5)

$$L_{\text{identity}} \leftarrow \frac{1}{m} \sum_{i=1}^m |G(x^{(i)}) - x^{(i)}| + \frac{1}{m} \sum_{i=1}^m |F(y^{(i)}) - y^{(i)}|.$$

8: Calculate the loss function.  $\triangleright$  Eq.(1)

$$L \leftarrow L_{\text{GAN}} + \lambda_1 L_{\text{cyc}} + \lambda_2 L_{\text{identity}}.$$

9: Calculate the gradients of loss function.

$$g_G \leftarrow \nabla_{\theta_G} L; g_F \leftarrow \nabla_{\theta_F} L.$$

$$g_X \leftarrow \nabla_{w_X} L; g_Y \leftarrow \nabla_{w_Y} L.$$

10: Minimize the loss function, using the Adam optimizer, with respect to the parameters of generators  $G, F$ .

$$\theta_G \leftarrow \theta_G - \alpha \cdot \text{Adam}(\theta_G, g_G); \theta_F \leftarrow \theta_F - \alpha \cdot \text{Adam}(\theta_F, g_F).$$

11: Maximize the loss function, using the Adam optimizer, with respect to the parameters of discriminators  $D_X, D_Y$ .

$$w_X \leftarrow w_X + \alpha \cdot \text{Adam}(w_X, g_X); w_Y \leftarrow w_Y + \alpha \cdot \text{Adam}(w_Y, g_Y).$$

12: **end while**

13: Decode  $G(X_{\text{test}})$  and  $F(Y_{\text{test}})$  with the JT-VAE decoder to obtain the molecules returned by the Mol-CycleGAN.

## Workflow

We conduct experiments to test whether the proposed model is able to generate molecules that possess desired properties and are close to the starting molecules. Namely, we evaluate the model on tasks related to *structural modifications*, as well as on tasks related to *molecule optimization*. For testing molecule optimization, we select the octanol-water partition coefficient (logP) penalized by the synthetic accessibility (SA) score and activity towards DRD2 receptor.

logP describes lipophilicity—a parameter influencing a whole set of other characteristics of compounds such as solubility, permeability through biological membranes, ADME (absorption, distribution, metabolism, and excretion) properties, and toxicity. We use the formulation as reported in the paper on JT-VAE [30], i.e. for molecule  $m$  the penalized logP is given as  $\log P(m) - SA(m)$ . We use the ZINC-250K dataset used in similar studies [19, 30], which contains 250 000 drug-like molecules extracted from the ZINC database [43].

For DRD2 activity task we use Random Forest classification model trained on ECFP fingerprints as the activity estimator (ROC AUC = 0.92), where the activity data were extracted from the ChEMBL database.

The detailed formulation of the tasks is the following:

- Structural transformations: We test the model's ability to perform simple structural transformations of the molecules. To this end, we choose the sets  $X$  and  $Y$ , differing in some structural aspects, and then test if our model can learn the transformation rules and apply them to molecules previously unseen by the model. These are the features by which we divide the sets:
  - Halogen moieties: We split the dataset into two subsets  $X$  and  $Y$ . The set  $Y$  consists of molecules which contain at least one of the following SMARTS: '[!#1]Cl', '[!#1]F', '[!#1]I', 'C#N', whereas the set  $X$  consists of such molecules which do not contain any of them. The SMARTS chosen in this experiment indicate halogen moieties and the nitrile group. Their presence and position within a molecule can have an immense impact on the compound's activity.
  - Bioisosteres: Molecules in set  $X$  are molecules with 'CN' and without 'CF<sub>3</sub>' group. The set  $Y$  consists of molecules which contain 'CF<sub>3</sub>' and does not contain 'CN' group.
  - CF<sub>3</sub> addition: The set  $X$  is a random sample from ZINC-250K (without 'CF<sub>3</sub>'). The set  $Y$  consists of molecules which contain 'CF<sub>3</sub>' group. This task is used as a control task for the bioisosteric substitu-



tion to check if the model can learn to generate this group at any position.

- Aromatic rings: Molecules in  $X$  have exactly two aromatic rings, whereas molecules in  $Y$  have one or three aromatic rings.
- Constrained molecule optimization: We optimize penalized logP, while constraining the degree of deviation from the starting molecule. The similarity between molecules is measured with Tanimoto similarity on Morgan Fingerprints [44]. The sets  $X$  and  $Y$  are random samples from ZINC-250K, where the compounds' penalized logP values are below and above the median, respectively.
- Unconstrained molecule optimization: We perform unconstrained optimization of penalized logP. The set  $X$  is a random sample from ZINC-250K and the set  $Y$  is a random sample from the top-20% molecules with the highest penalized logP in ZINC-250K.
- Activity: We use the Mol-CycleGAN to create active molecules from inactive ones, where DRD2 (dopamine receptor D2) was chosen as the biological target. Compounds with annotated activity towards the target were extracted from ChEMBL database, version 25 [45]. We split the dataset into two subsets, active ( $Y$ ) and inactive ( $X$ ). The set  $Y$  consists of molecules with  $K_i < 100$ , whereas all remaining molecules are delegated to set  $X$ .

### Composition of the datasets

**Dataset sizes** In Tables 1 and 2 we show the number of molecules in the datasets used for training and testing. In all experiments we use separate sets for training the

**Table 1 Structural transformations—dataset sizes**

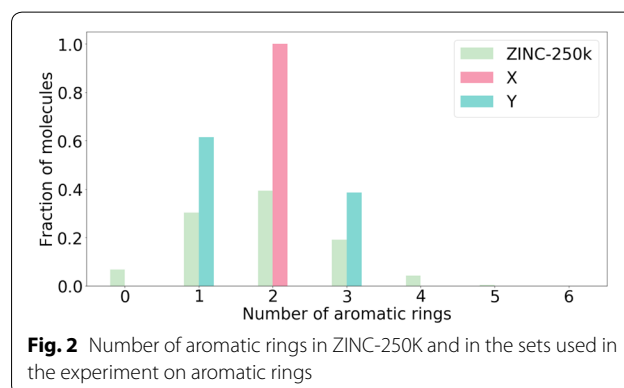
Dataset	Halogen moieties	Aromatic rings	Bioisosteres	CF <sub>3</sub>
$X_{\text{train}}$	75,000	80,000	12,454	20,000
$X_{\text{test}}$	86,899	18,220	800	800
$Y_{\text{train}}$	75,000	80,000	8321	8799
$Y_{\text{test}}$	12,556	43,193	800	–

**Table 2 Physicochemical transformations—dataset sizes**

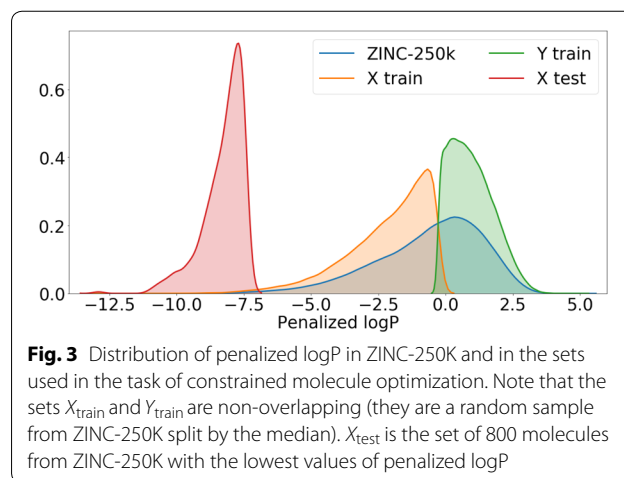
Dataset	Constrained optimization	Unconstrained optimization	Activity
$X_{\text{train}}$	80,000	80,000	4500
$X_{\text{test}}$	800	800	333
$Y_{\text{train}}$	80,000	24,946	2511
$Y_{\text{test}}$	–	–	–

model ( $X_{\text{train}}$  and  $Y_{\text{train}}$ ) and separate, non-overlapping ones for evaluating the model ( $X_{\text{test}}$  and  $Y_{\text{test}}$ ). In CF<sub>3</sub> addition and all physicochemical experiments no  $Y_{\text{test}}$  set is required.

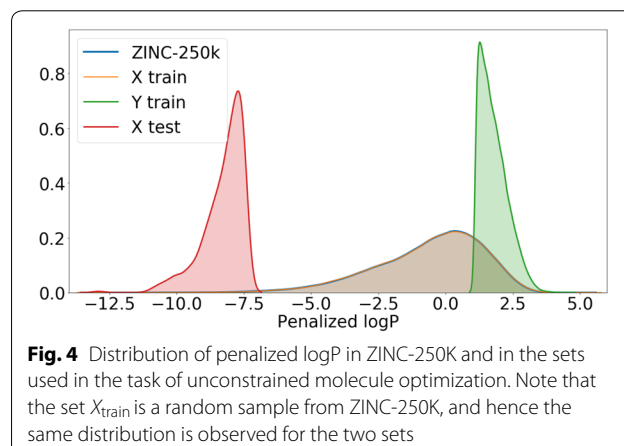
**Distribution of the selected properties** In the experiment on halogen moieties, the set  $X$  always (i.e., both in train- and test-time) contains molecules without halogen



**Fig. 2** Number of aromatic rings in ZINC-250K and in the sets used in the experiment on aromatic rings



**Fig. 3** Distribution of penalized logP in ZINC-250K and in the sets used in the task of constrained molecule optimization. Note that the sets  $X_{\text{train}}$  and  $Y_{\text{train}}$  are non-overlapping (they are a random sample from ZINC-250K split by the median).  $X_{\text{test}}$  is the set of 800 molecules from ZINC-250K with the lowest values of penalized logP



**Fig. 4** Distribution of penalized logP in ZINC-250K and in the sets used in the task of unconstrained molecule optimization. Note that the set  $X_{\text{train}}$  is a random sample from ZINC-250K, and hence the same distribution is observed for the two sets

**Table 3** Evaluation of models modifying the presence of halogen moieties and the number of aromatic rings

	Halogen moieties		Aromatic rings	
	$X \rightarrow G(X)$	$Y \rightarrow F(Y)$	$X \rightarrow G(X)$	$Y \rightarrow F(Y)$
Success rate	0.6429	0.7161	0.5342	0.4216
Non-identity	0.9345	0.9574	0.9082	0.8899
Uniqueness	0.9952	0.9953	0.9957	0.9954

Success rate is the fraction of times when a desired modification occurs. Non-identity is the fraction of times when the generated molecule is different from the starting one. Uniqueness is the fraction of unique molecules in the set of generated molecules

moieties, and the set  $Y$  always contains molecules with halogen moieties. In the dataset used to construct the latent space (ZINC-250K) 65% molecules do not contain any halogen moiety, whereas the remaining 35% contain one or more halogen moieties.

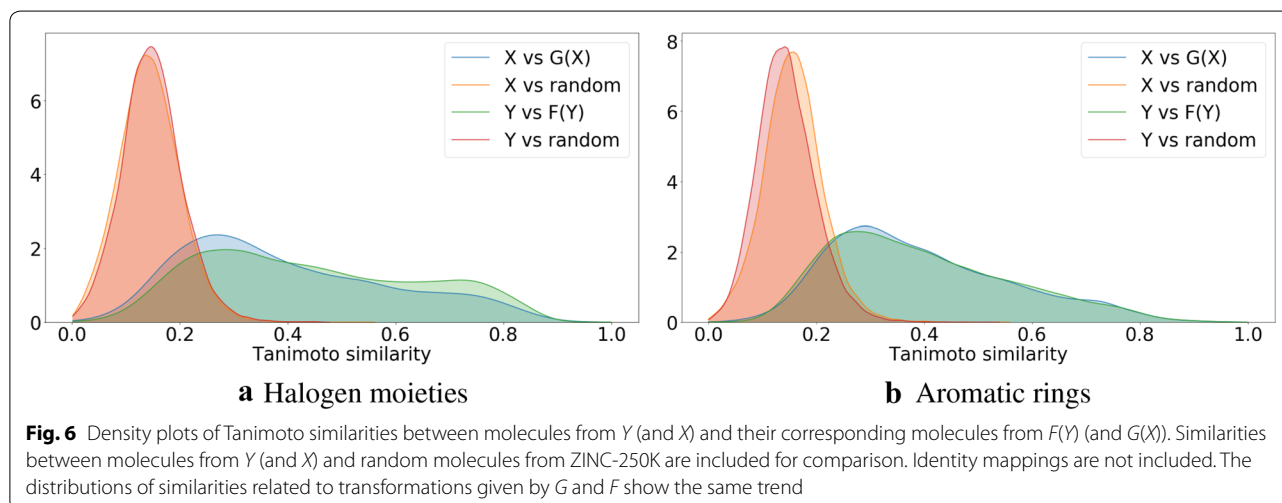
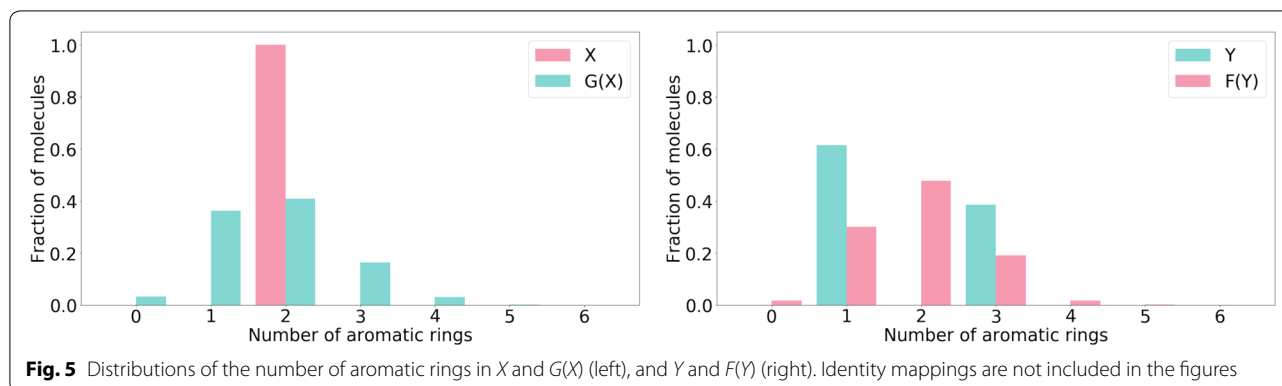
In the experiment on aromatic rings, the set  $X$  always (i.e., both in train- and test-time) contains molecules with 2 rings, and the set  $Y$  always contains molecules with 1 or 3 rings. The distribution of the number of aromatic rings

in the dataset used to construct the latent space (ZINC-250K) is shown in Fig. 2 along with the distribution for  $X$  and  $Y$ .

In the bioisosteres experiment, the set  $X$  always contains molecules with CN group and without  $CF_3$  group. Set  $Y$  always contains molecules with  $CF_3$  group. In the  $CF_3$  addition experiment, the set  $X$  is a random sample from ZINC-250K, and the set  $Y$  similarly contains molecules with  $CF_3$  group. In the dataset used to construct the latent space (ZINC-250K) 5.1% of molecules contain CN group, whereas molecules with  $CF_3$  group accounts for 3.8% of total dataset.

For the molecule optimization tasks we plot the distribution of the property being optimized (penalized logP) in Fig. 3 (constrained optimization) and Fig. 4 (unconstrained optimization).

In the activity optimization experiment, the set  $X$  contains inactive molecules and the set  $Y$  contains active molecules. The mean activity prediction equals 0.223 for the whole dataset which was used to construct the latent space (ZINC-250K), whereas for the  $X_{\text{test}}$  dataset the mean predicted activity is 0.179.



### Architecture of the models

All networks are trained using the Adam optimizer [46] with learning rate 0.0001. During training we use batch normalization [47]. As the activation function we use leaky-ReLU with  $\alpha = 0.1$ . In the structural experiments the models are trained for 100 epochs and in the physicochemical experiments for 300 epochs.

### Structural data experiments

- *Generators* are built of one fully connected residual layer, followed by one dense layer. All layers contain 56 units.
- *Discriminators* are built of 6 dense layers of the following sizes: 56, 42, 28, 14, 7, 1 units.

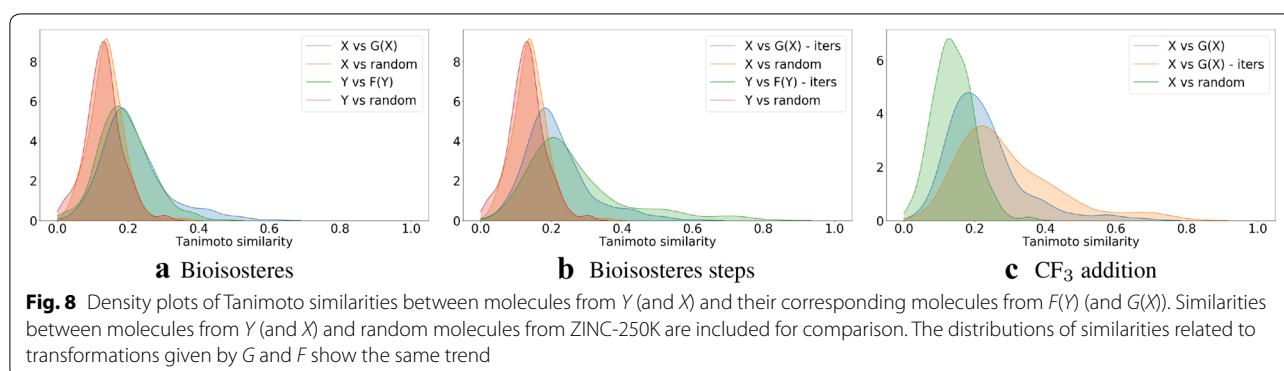
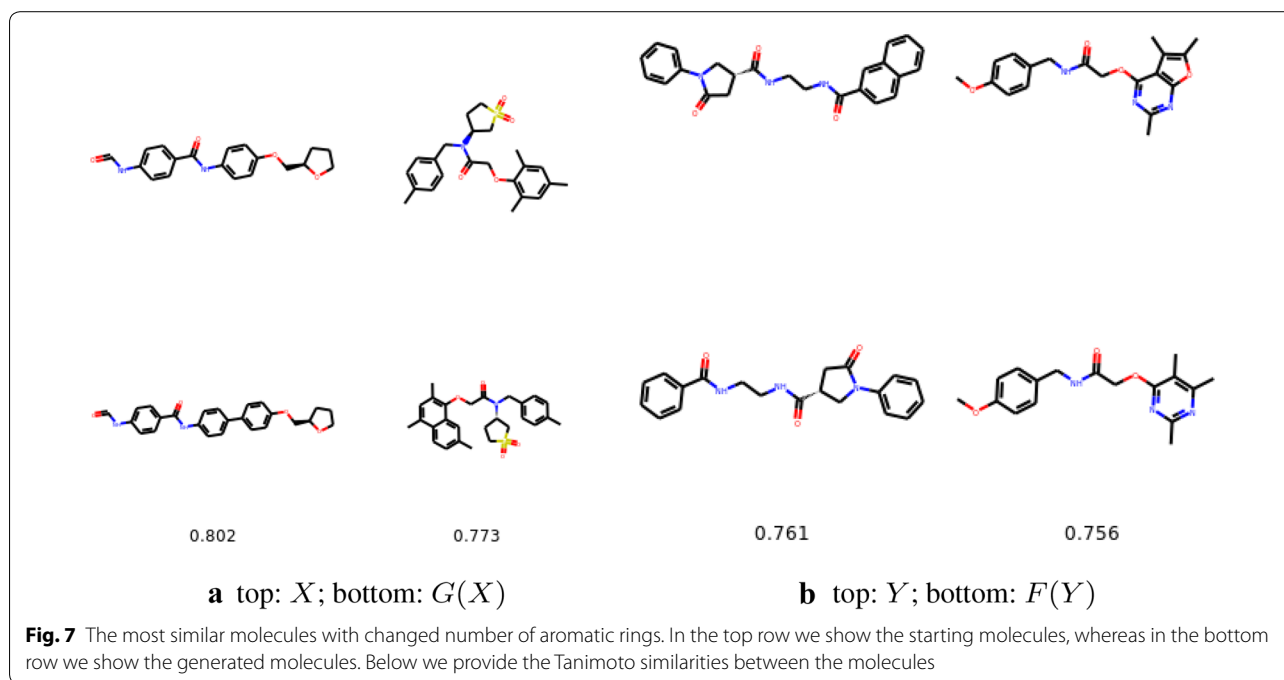
### Physicochemical data experiments

- *Generators* are built of four fully connected residual layers. All layers contain 56 units.
- *Discriminators* are built of 7 dense layers of the following sizes: 48, 36, 28, 18, 12, 7, 1 units.

### Results and discussion

#### Structural transformations

In each structural experiment we test the model's ability to perform simple transformations of molecules in both directions  $X \rightarrow Y$  and  $Y \rightarrow X$ . Here,  $X$  and  $Y$  are non-overlapping sets of molecules with a specific structural property. We start with experiments on structural properties because they are easier to interpret and the rules



**Table 4 Evaluation of models performing the bioisosteric substitution**

	Single step		10 intermediate steps	
	$X \rightarrow G(X)$	$Y \rightarrow F(Y)$	$X \rightarrow G(X)$	$Y \rightarrow F(Y)$
Success rate	0.216	0.450	0.514	0.776
Diversity	0.998	0.996	0.998	1.000
Non-identity	1.000	1.000	0.596	0.816

**Table 5 Evaluation of models modifying the presence of CF<sub>3</sub> group**

	Single step	10 intermediate steps
	Success rate	0.191
Diversity	0.995	1.000
Non-identity	1.000	0.410

related to transforming between  $X$  and  $Y$  are well defined. Hence, the present task should be easier for the model, as compared to the optimization of complex molecular properties, for which there are no simple rules connecting  $X$  and  $Y$ .

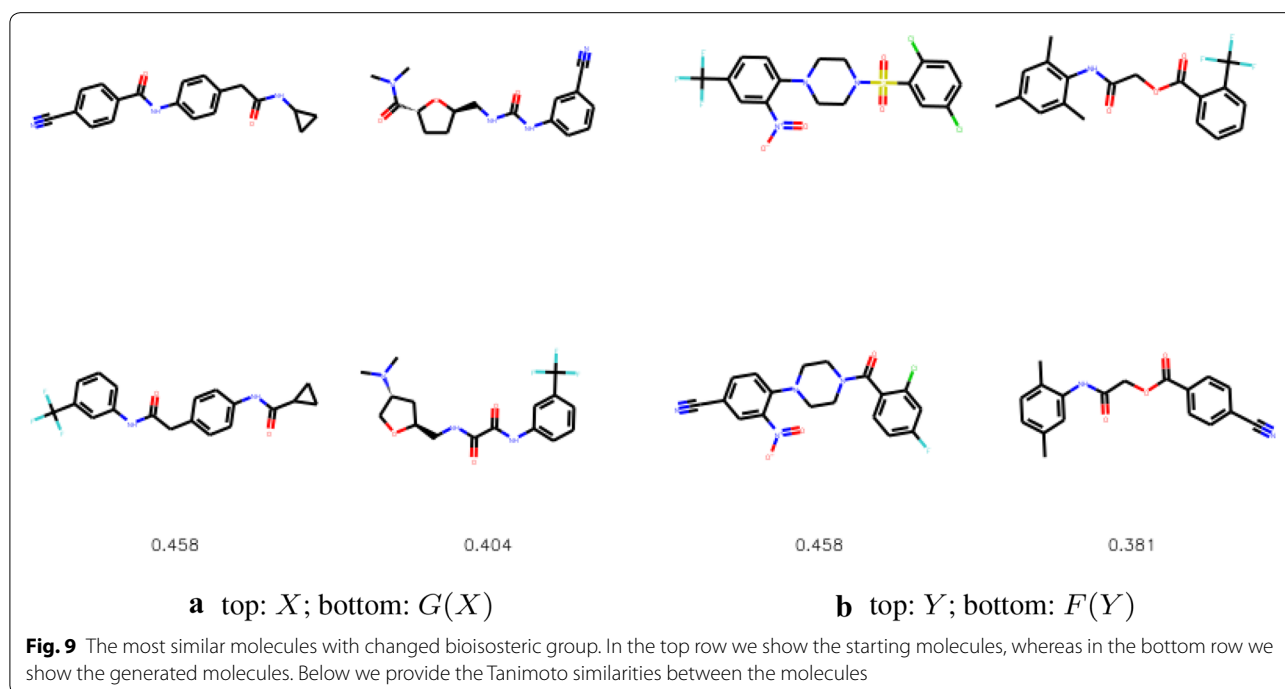
In Table 3 we show the success rates for the tasks of performing structural transformations of molecules. The task of changing the number of aromatic rings is more difficult than changing the presence of halogen moieties.

In the former the transition between  $X$  (with 2 rings) and  $Y$  (with 1 or 3 rings, cf. Fig. 5) is more than a simple addition/removal transformation, as it is in the other case (see Fig. 5 for the distributions of the aromatic rings). This is reflected in the success rates which are higher for the task of transformations of halogen moieties. In the dataset used to construct the latent space (ZINC-250K) 64.9% molecules do not contain any halogen moiety, whereas the remaining 35.1% contain one or more halogen moieties. This imbalance might be the reason for the higher success rate in the task of removing halogen moieties ( $Y \rightarrow F(Y)$ ). Molecular similarity and drug-likeness are achieved in all experiments.

To confirm that the generated molecules are close to the starting ones, we show in Fig. 6 distributions of their Tanimoto similarities (using Morgan fingerprints). For comparison we also include distributions of the Tanimoto similarities between the starting molecule and a random molecule from the ZINC-250K dataset. The high similarities between the generated and the starting molecules show that our procedure is neither a random sampling from the latent space nor a memorization of the manifold in the latent space with the desired value of the property. In Fig. 7 we visualize the molecules, which after transformation are the most similar to the starting molecules.

#### Bioisosteres

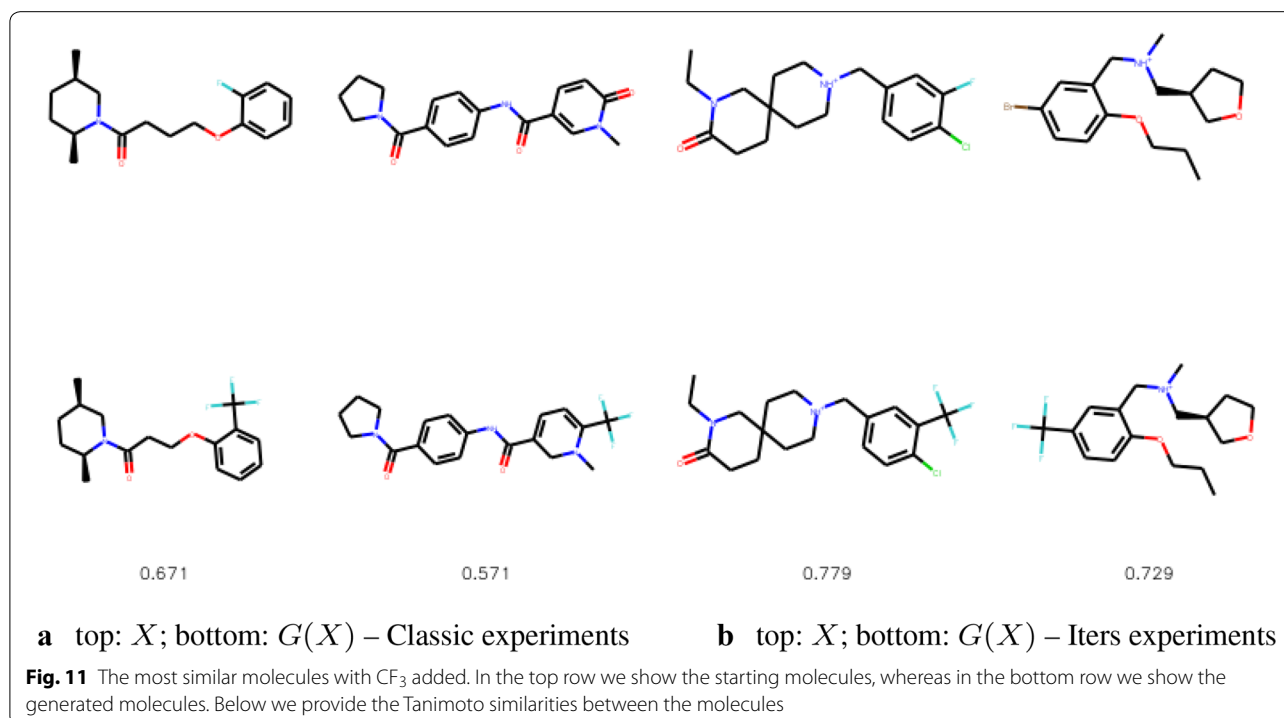
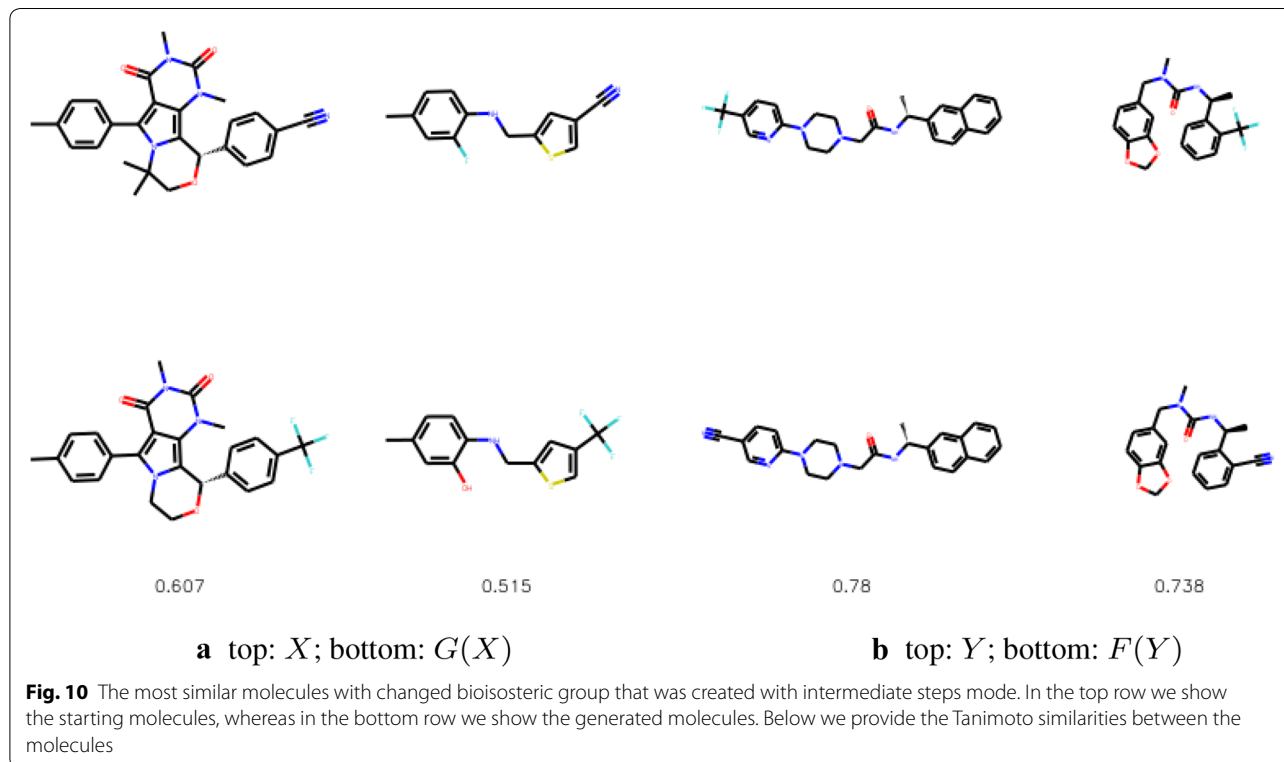
As for the more complicated structural transformation, we present a bioisosteric substitution task. Here, we





have sets  $X$  and  $Y$  with groups CN and  $CF_3$  respectively. These two moieties have similar electronic effects, CN being more hydrophilic. The dataset was constructed so that there are no compounds containing both of these

fragments at once. We want to see whether our method can learn to substitute one group with another, or it will put the target group at a random position in the molecule.



Three different optimization procedures are performed: (a) bioisosteric substitution conducted as described above, (b) generating 10 intermediate steps from the bioisosteric substitution optimization path ( $x, G(x)$ ), and (c) the addition of  $\text{CF}_3$  group. In the step wise variant, molecules from the optimization path were taken in equal intervals. In the case of  $\text{CF}_3$  addition, we use  $X$  without trifluoromethyl group and  $Y$  with the group present within the structure. Here, similarly as in the halogen example, we check if our model can learn to include the given substructure in the generated molecule. Here, we treat the  $\text{CF}_3$  addition task as a control task for the bioisosteric substitution since it should be easier for the model to add the group in some indefinite position. Figure 8 shows similarities between original and optimized datasets in these three experiments. The plots show that this time the trained transformation leads to

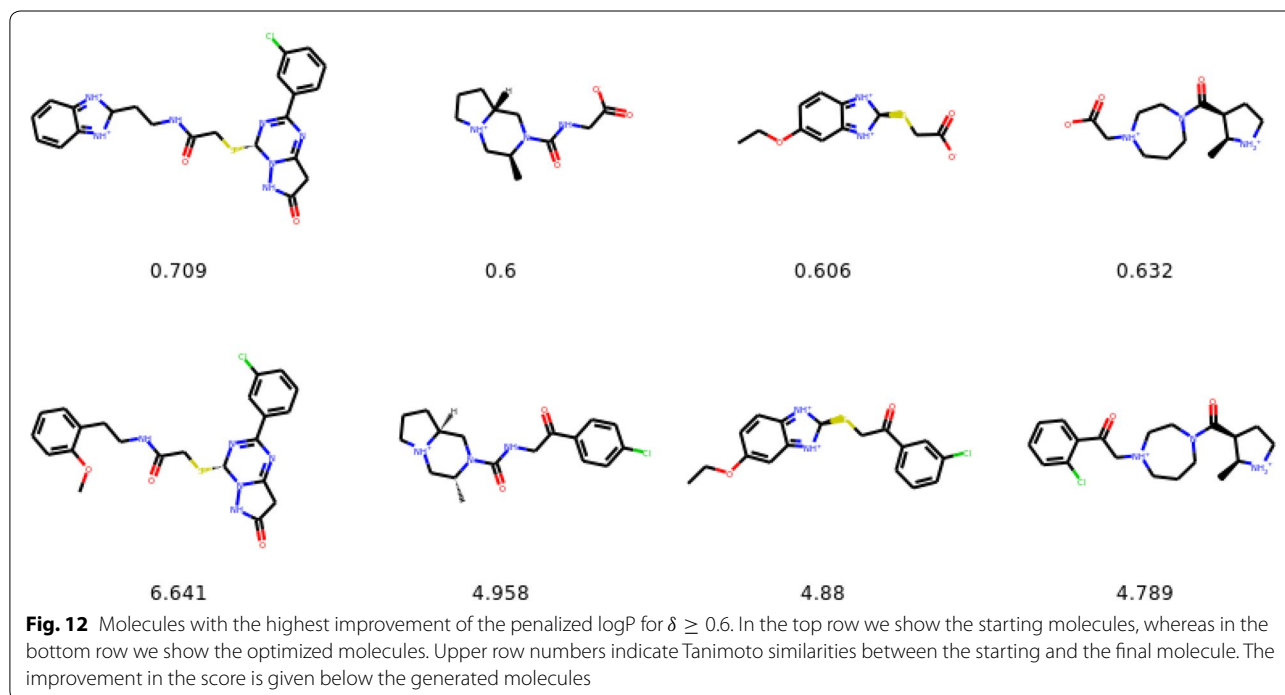
more dissimilar molecules, which is probably caused by two major changes in the structure of a compound—first we remove one group, and then we add another group. Comparing similarity distribution to our control task of trifluoromethyl group addition, the latter leads to greater similarity of the generated compounds.

Tables 4 and 5 summarize quantitatively the results of bioisosteric substitution. All the generated molecules maintain high diversity. Interestingly, inverse optimization (substitution of  $\text{CF}_3$  group with CN) is an easier task. The reason behind that is probably that  $\text{CF}_3$  fragment contains more atoms, and thus its decoding process is more complex. Moreover, it appears that addition of the  $\text{CF}_3$  group is a more difficult task than substitution as the success rate is lower here. The higher rates in the substitution variant may be caused by high similarity of two datasets  $X$  and  $Y$ , which both consist of molecules

**Table 6 Results of the constrained optimization for Junction Tree Variational Autoencoder [30] (JT-VAE), Graph Convolutional Policy Network [35] (GCPN) and Mol-CycleGAN**

$\delta$	JT-VAE			GCPN			Mol-CycleGAN		
	Improvement	Similarity	Success (%)	Improvement	Similarity	Success (%)	Improvement	Similarity	Success (%)
0	1.91 ± 2.04	0.28 ± 0.15	97.5	4.20 ± 1.28	0.32 ± 0.12	100.0	8.30 ± 1.98	0.16 ± 0.09	99.75
0.2	1.68 ± 1.85	0.33 ± 0.13	97.1	4.12 ± 1.19	0.34 ± 0.11	100.0	5.79 ± 2.35	0.30 ± 0.11	93.75
0.4	0.84 ± 1.45	0.51 ± 0.10	83.6	2.49 ± 1.30	0.47 ± 0.08	100.0	2.89 ± 2.08	0.52 ± 0.10	58.75
0.6	0.21 ± 0.75	0.69 ± 0.06	46.4	0.79 ± 0.63	0.68 ± 0.08	100.0	1.22 ± 1.48	0.69 ± 0.07	19.25

The biggest improvements across all methods are italicized



with one of the two groups with a similar bioactivity effect.

We compare the substituted compounds qualitatively in Figs. 9 and 10. We observe that the moieties are often correctly substituted with only minor changes to the overall compound structure. The method learns to substitute bioisosteric groups rather than attach the new group to other fragments of a molecule. Figure 11 shows the addition scenario, in which again changes to the molecule are small. Additionally,  $\text{CF}_3$  group also prefers replacing other atoms, e.g. halogen groups or ketone groups in the examples provided.

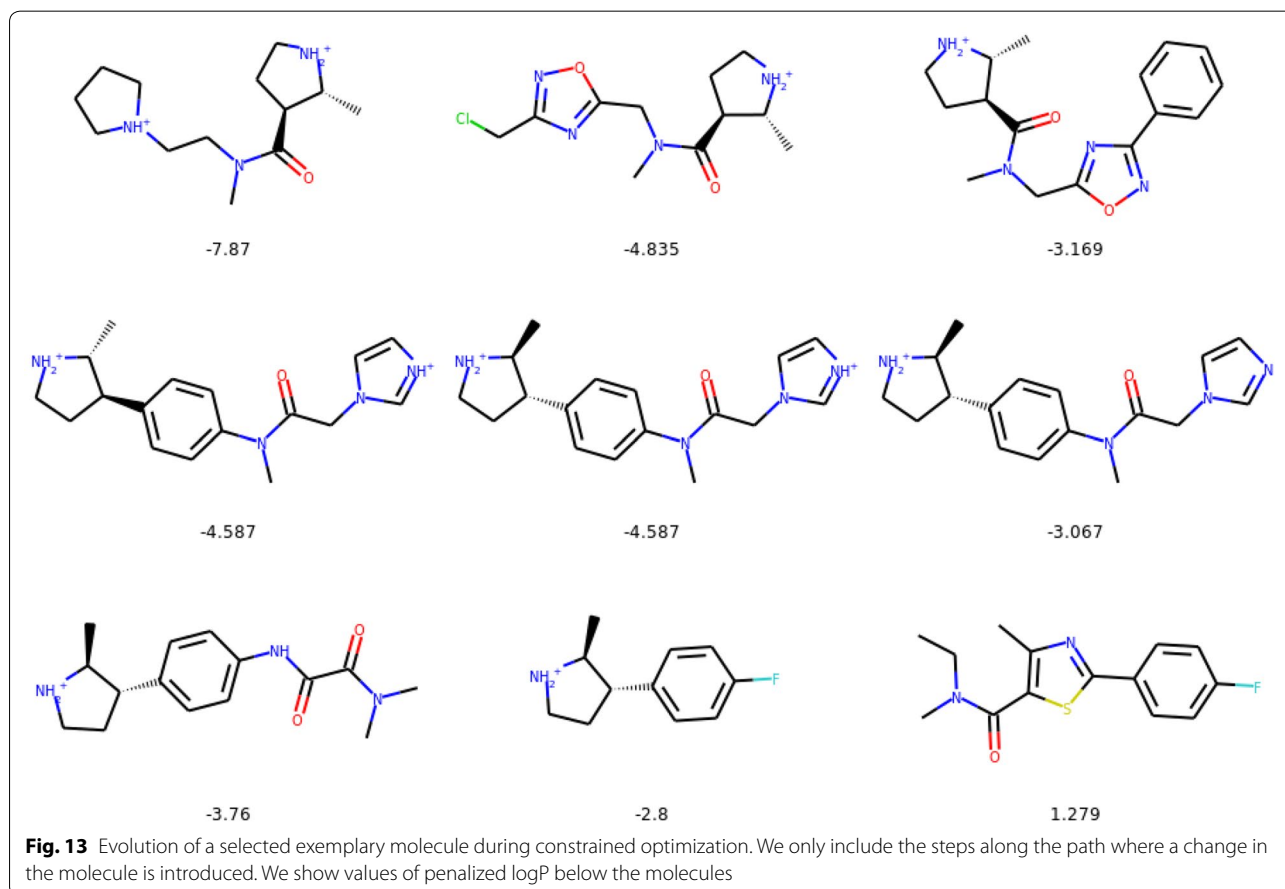
### Constrained molecule optimization

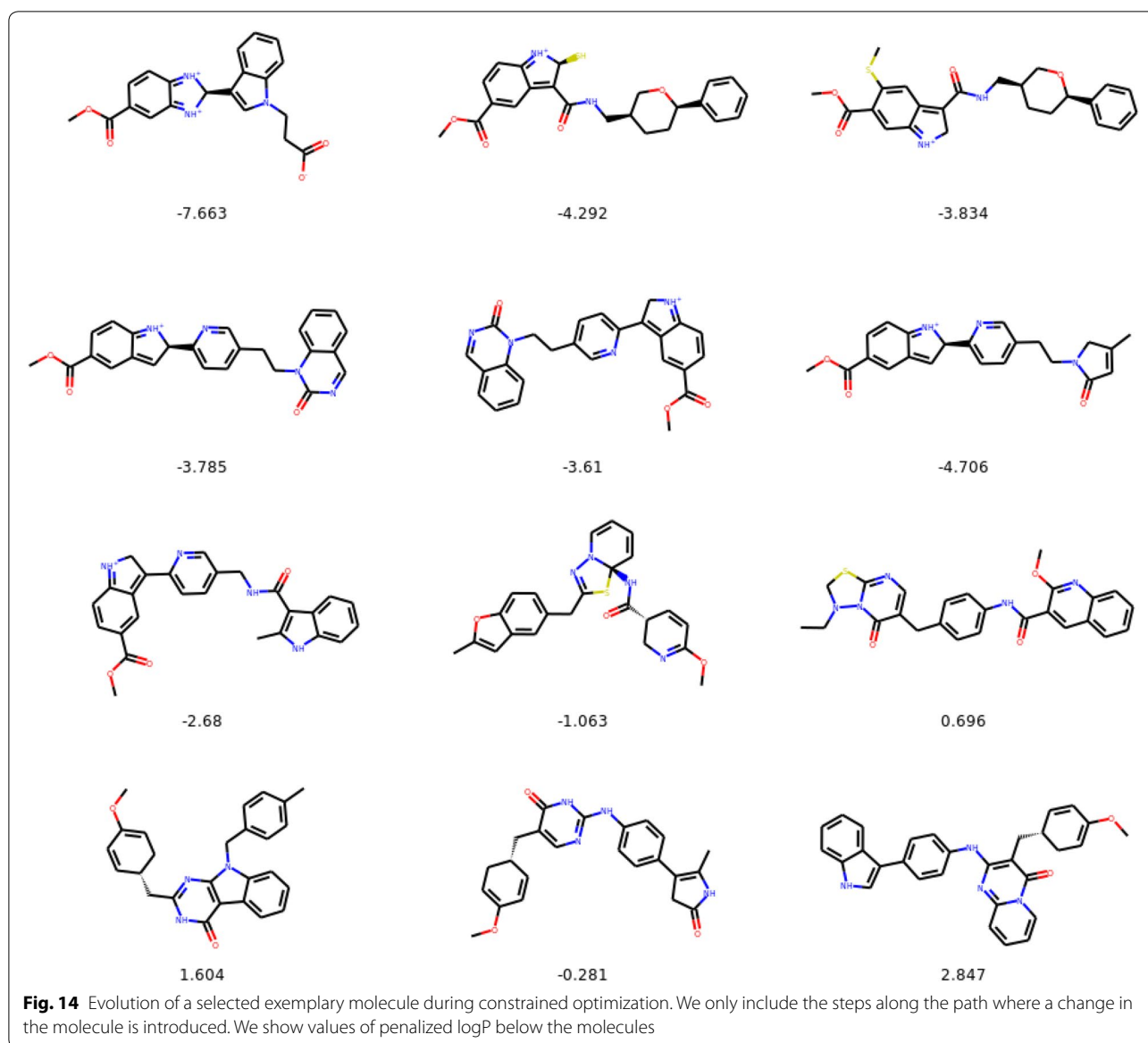
As our main task we optimize the desired property under the constraint that the similarity between the original and the generated molecule is higher than a fixed threshold (denoted as  $\delta$ ). This is a more realistic scenario in drug discovery, where the development of new drugs usually starts with known molecules such as existing drugs [48]. Here, we maximize the penalized logP coefficient and use the Tanimoto similarity with the Morgan fingerprint [44] to define the threshold of similarity,  $\text{sim}(m, m') \geq \delta$ . We compare our results with previous similar studies [30, 35].

In our optimization procedure each molecule (given by the latent space coordinates  $x$ ) is fed into the generator to obtain the 'optimized' molecule  $G(x)$ . The pair  $(x, G(x))$  defines what we call an 'optimization path' in the latent space of JT-VAE. To be able to make a comparison with the previous research [30], we start the procedure from the 800 molecules with the lowest values of penalized logP in ZINC-250K, and then we decode molecules from  $K = 80$  points along the path from  $x$  to  $G(x)$  in equal steps.

From the resulting set of molecules we report the molecule with the highest penalized logP score that satisfies the similarity constraint. A modification succeeds if one of the decoded molecules satisfies the constraint and is distinct from the starting one. Figure 12 shows exemplary molecules with highest improvements and high similarity to the starting compounds.

In the task of optimizing penalized logP of *drug-like* molecules, our method significantly outperforms the previous results in the mean improvement of the property (see Table 6). It achieves a comparable mean similarity in the constrained scenario (for  $\delta > 0$ ). The success rates are comparable for  $\delta = 0, 0.2$ , whereas for the more stringent constraints ( $\delta = 0.4, 0.6$ ) our model has lower success rates.





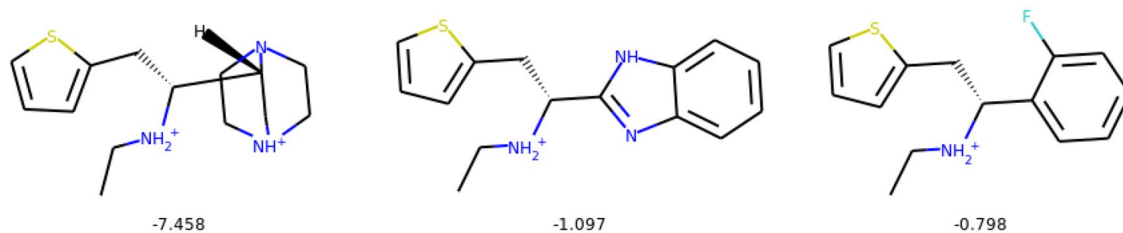
Note that comparably high improvements of penalized logP can be obtained using reinforcement learning [35]. However, many methods using reinforcement learning tend to generate compounds that are not drug-like because they suffer from catastrophic forgetting when the optimization task is changed, e.g. they learn the prior drug-like distribution first, and then they try to increase the logP property at the cost of divergence from the prior distribution. Nonetheless, this problem can be relatively easily alleviated, e.g., by multi-target optimization that takes QED [49] into account. In our method (as well as in JT-VAE) drug-likeness is achieved “by design” and is an intrinsic feature of the latent space obtained by training the variational autoencoder on molecules from ZINC (which are drug-like).

#### Molecular paths from constrained optimization experiments

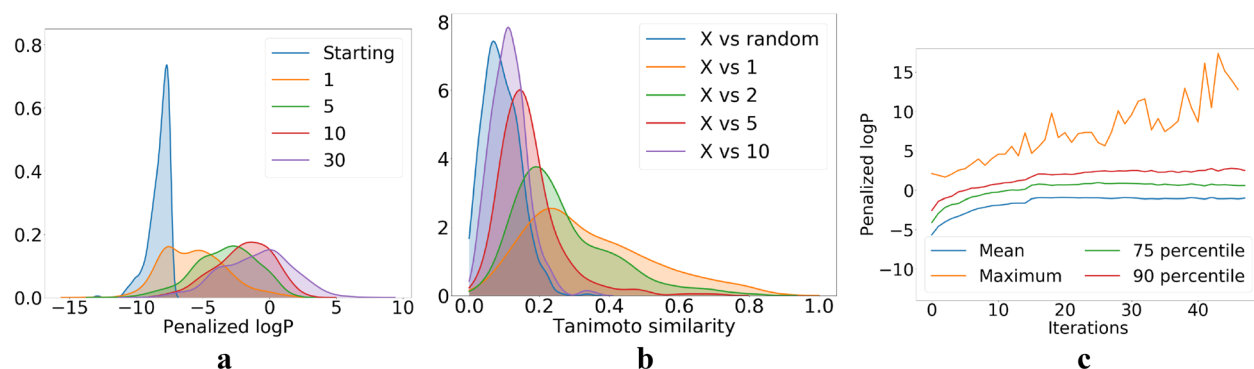
In the following section we show examples of the evolution of the selected molecules for the constrained optimization experiments. Figures 13, 14, and 15 show starting and final molecules, together with all molecules generated along the optimization path, and their values of penalized logP.

#### Unconstrained molecule optimization

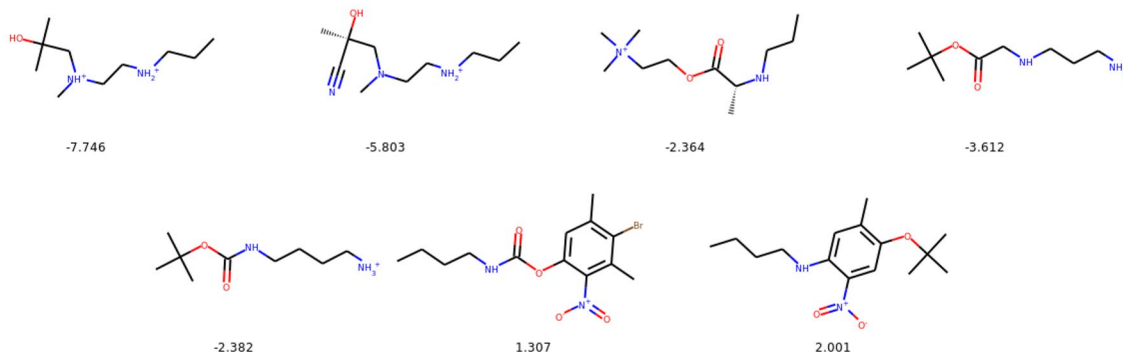
Our architecture is tailor-made for the scenario of constrained molecule optimization. However, as an additional task, we check what happens when we iteratively use the generator on the molecules being optimized. This should lead to diminishing similarity between the starting molecules and those in consecutive iterations. For



**Fig. 15** Evolution of a selected exemplary molecule during constrained optimization. We only include the steps along the path where a change in the molecule is introduced. We show values of penalized logP below the molecules



**Fig. 16** Results of iterative procedure of the unconstrained optimization. **a** Distribution of the penalized logP in the starting set and after  $K = 1, 5, 10, 30$  iterations. **b** Distribution of the Tanimoto similarity between the starting molecules  $X$  and random molecules from ZINC-250K, as well as those generated after  $K = 1, 2, 5, 10$  iterations. **c** Plot of the mean value, percentiles (75th and 90th), and the maximum value of penalized logP as a function of the number of iterations



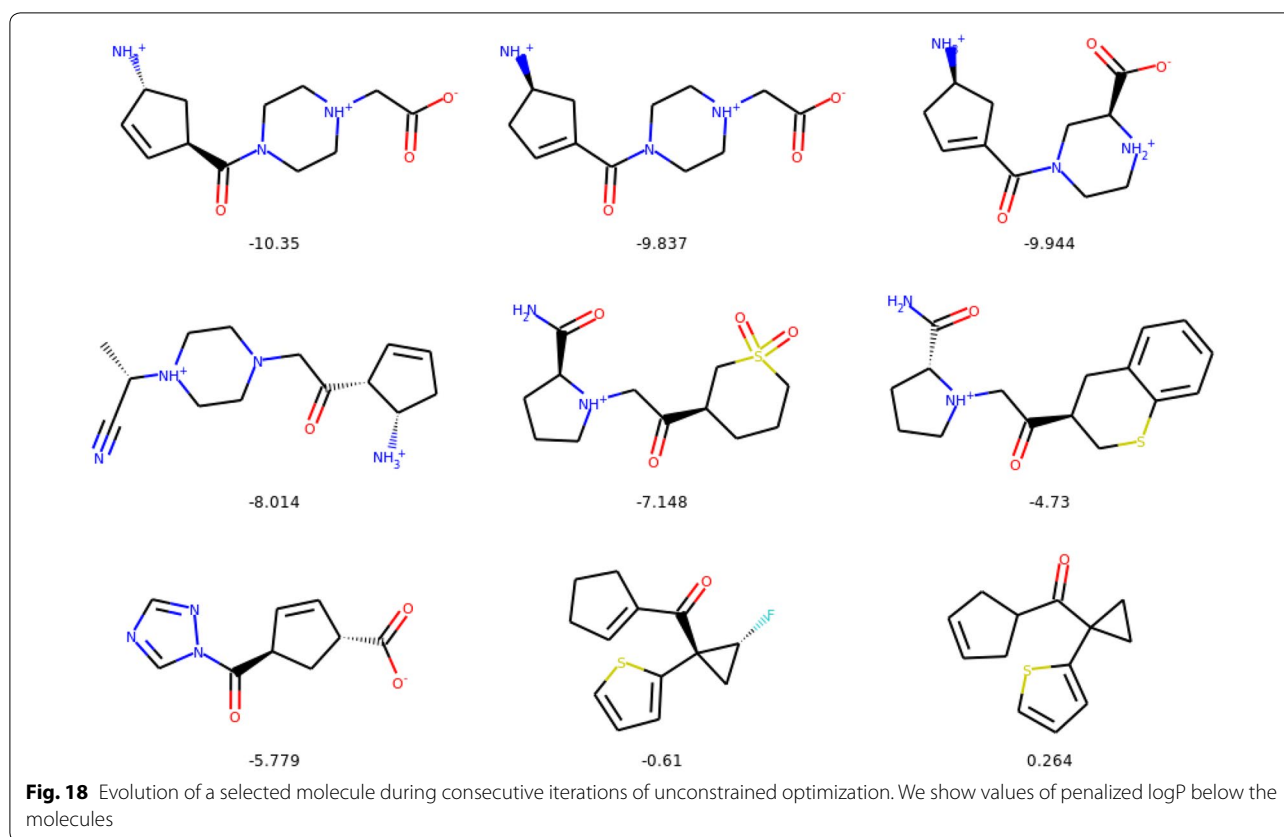
**Fig. 17** Evolution of a selected molecule during consecutive iterations of unconstrained optimization. We show values of penalized logP below the molecules

the present task the set  $X$  needs to be a sample from the entire ZINC-250K, whereas the set  $Y$  is chosen as a sample from the top-20% of molecules with the highest value of penalized logP. Each molecule is fed into the generator and the corresponding ‘optimized’ molecule’s latent space representation is obtained. The generated latent space representation is then treated as the new input for the generator. The process is repeated  $K$  times and the

resulting set of molecules is  $\{G(x), G(G(x)), \dots\}$ . Here, as in the previous task and as in previous research [30] we start the procedure from the 800 molecules with the lowest values of penalized logP in ZINC-250K.

The results of our unconstrained molecule optimization are shown in Fig. 16. In Fig. 16a, c we observe that consecutive iterations keep shifting the distribution of the objective (penalized logP) towards higher values.





However, the improvement from further iterations is decreasing. Interestingly, the maximum of the distribution keeps increasing (although in somewhat random fashion). After 10–20 iterations it reaches very high values of logP observed from molecules which are not drug-like, similarly to those obtained with RL [35]. Both in the case of the RL approach and in our case, the molecules with the highest penalized logP after many iterations also become non-drug-like—see Fig. 19 for a list of compounds with the maximum values of penalized logP in the iterative optimization procedure. This lack of drug-likeness is related to the fact that after performing many iterations, the distribution of coordinates of our set of molecules in the latent space goes far away from the prior distribution (multivariate normal) used when training the JT-VAE on ZINC-250K. In Fig. 16b we show the evolution of the distribution of Tanimoto similarities between the starting molecules and those obtained after  $K = 1, 2, 5, 10$  iterations. We also show the similarity between the starting molecules and random molecules from ZINC-250K. We observe that after 10 iterations the similarity between the starting molecules and the optimized ones is comparable to the similarity of random molecules from ZINC-250K. After around 20 iterations the optimized molecules become less similar to the starting ones than random molecules from ZINC-250K, as

the set of optimized molecules is moving further away from the space of drug-like molecules.

#### **Molecular paths from unconstrained optimization experiments**

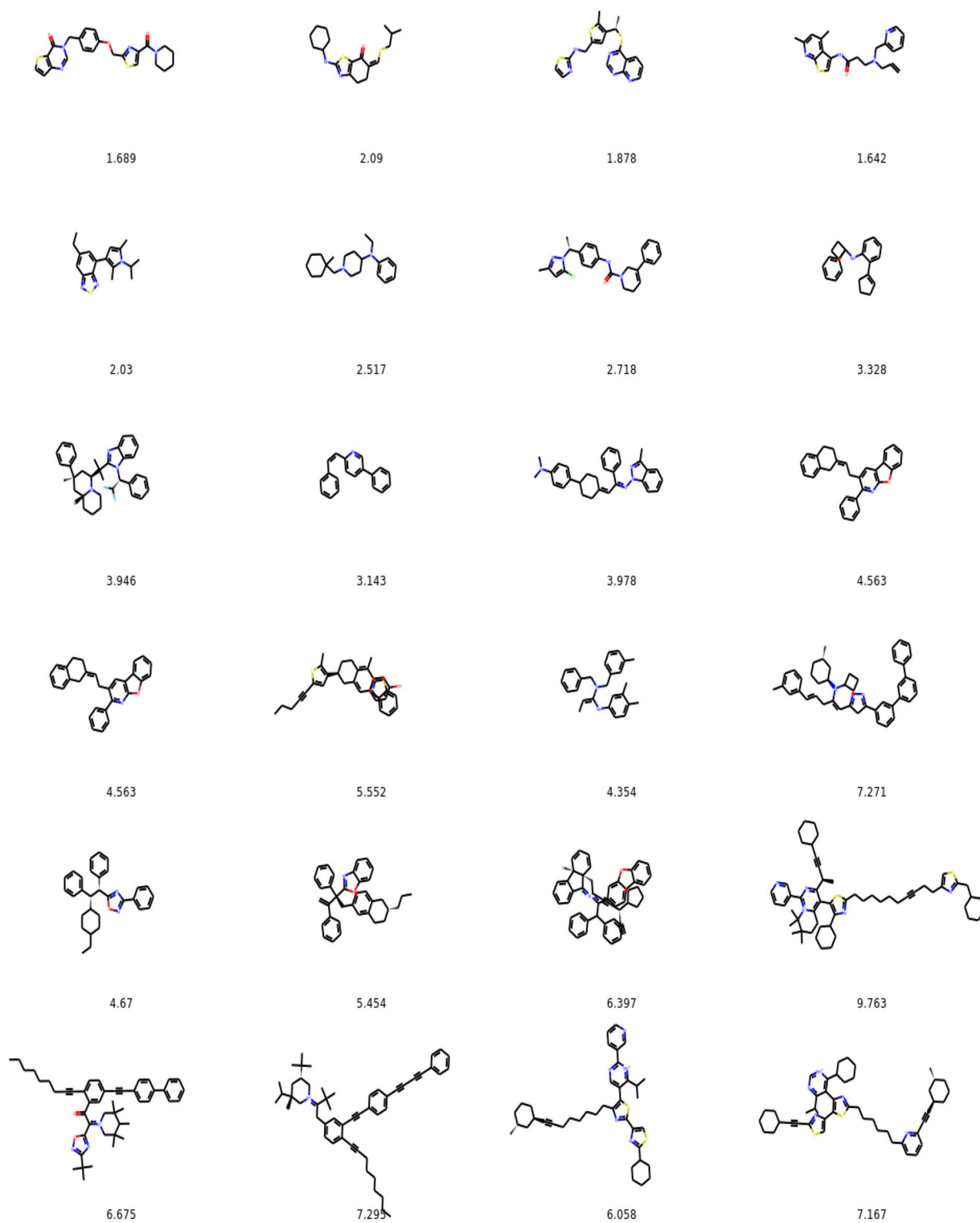
In the following section we show examples of the evolution of selected molecules for the unconstrained optimization experiments. Figures 17 and 18 show starting and final molecules, together with all molecules generated during the iteration over the optimization path and their penalized logP values.

#### **Molecules with the highest values of penalized logP**

On Fig. 16c we plot the maximum value of penalized logP in the set of molecules being optimized as a function of number of iterations for unconstrained molecule optimization. In Fig. 19 we show corresponding molecules for iterations 1–24.

#### **Activity**

Lastly, we test compound activity optimization for the dopamine receptor D2, i.e. we want to increase the binding affinity of a compound towards DRD2. For this task we selected a set  $X$  of inactive compounds, and a set  $Y$  of active molecules which were extracted from the ChEMBL database. We used threshold of  $K_i < 100$  nM



**Fig. 19** Molecules with the highest penalized logP in the set being optimized for iterations 1–24 for unconstrained optimization. We show values of penalized logP below the molecules

**Table 7** Quantitative evaluation of the compounds with optimized activity

Success rate	0.874
Diversity	0.994
Non-identity	1.000

**Table 8** Activity predictions and statistics for considered datasets

ZINC-250K predicted mean activity	0.223
$X_{\text{test}}$ predicted mean activity	0.179
$G(X_{\text{test}})$ predicted mean activity	0.362
Mean gain in predicted activity	0.182

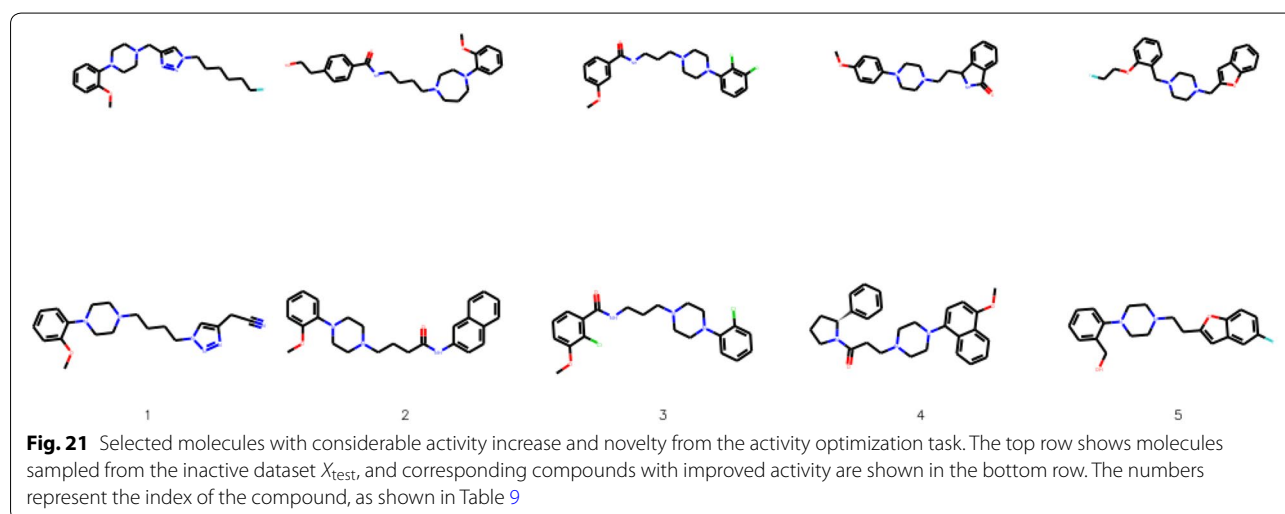
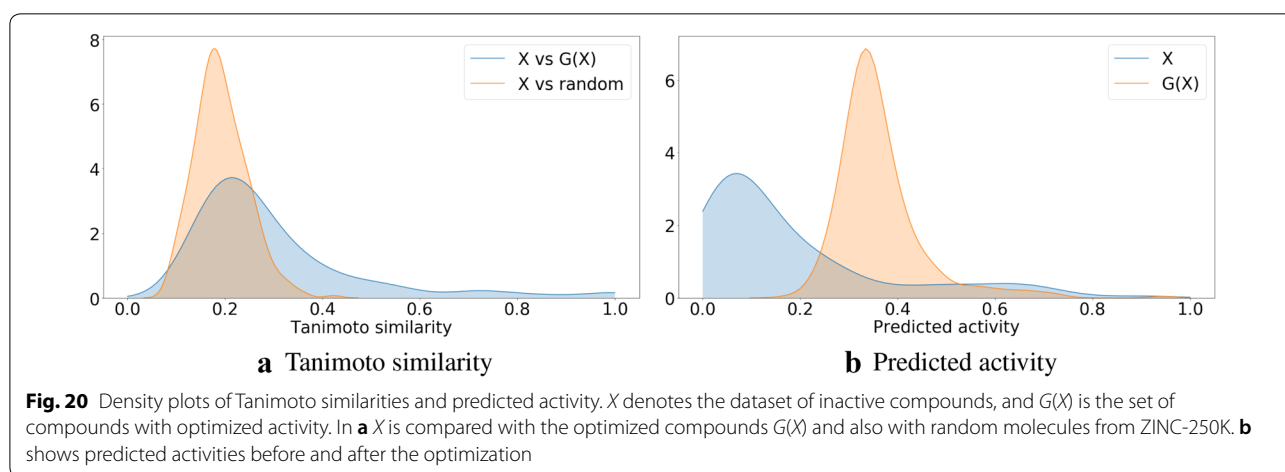
for selecting active compounds (2738 active compounds and 2254 inactive compounds were selected for training after filtering out duplicates).

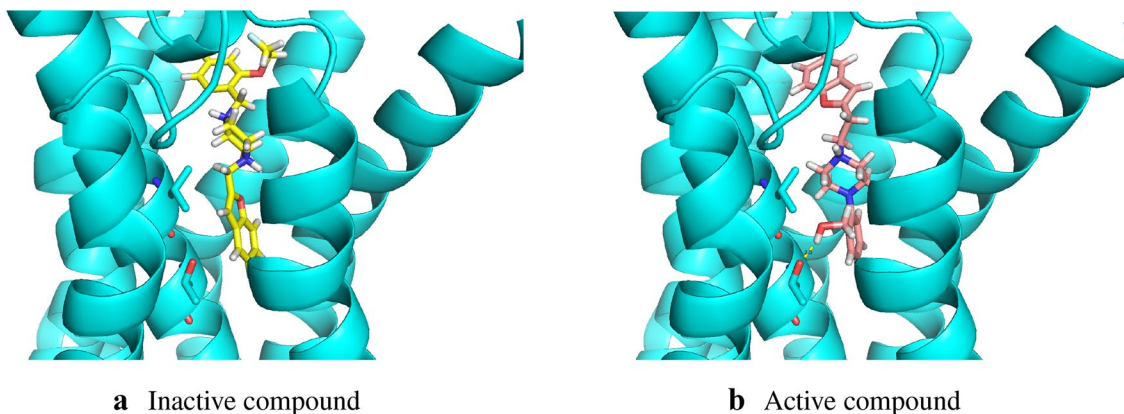
**Table 9** Statistics of the 5 optimized compounds presented in Fig. 21

Index	Activity original	Activity generated	$\Delta_{\text{Activity}}$	Tanimoto
1	0.32	0.71	0.39	0.703
2	0.23	0.73	0.50	0.496
3	0.16	0.49	0.33	0.702
4	0.03	0.52	0.49	0.395
5	0.05	0.47	0.42	0.458

Activity of the original inactive compound along with the predicted activity of the optimized compound is shown, as well as change in the predicted activity and Tanimoto similarity between each pair of compounds

For scoring the generated molecules, we trained a DRD2 activity prediction classification model based on ECFP fingerprints (generated with RDKit [50]). We chose to use a random forest model with 0.92 ROC AUC test score in threefold cross-validation. In this task we





**Fig. 22** Exemplary docking of a compound (index 5 in Table 9) and its optimized variant. We can see, that due to the removal of fluoroethyl group, the compound rotated by 180 degrees and was able to form additional hydrogen bond, stabilizing the complex. The docking energy was improved from  $-8.8$  (a) to  $-10.2$  kcal/mol (b)

also add 10 intermediate molecules from the optimization path to find more similar compound with improved activity. Table 7 quantitatively summarizes the experiment of activity optimization. Table 8 shows that the Mol-CycleGAN is able to increase activity of a selected inactive drug by a significant margin, based on the prediction of a bioactivity model. Figure 20 shows similarity of the optimized compounds to the starting molecules and compares their predicted activities. Examples of optimized compounds are presented in Fig. 21. To validate the results of the experiment, we performed docking procedures for a number of generated compounds and found that, on average, the optimized compounds have better docking energies than their progenitors (Fig. 22).

## Conclusions

In this work, we introduce Mol-CycleGAN—a new model based on CycleGAN which can be used for the de novo generation of molecules. The advantage of the proposed model is the ability to learn transformation rules from the sets of compounds with desired and undesired values of the considered property. The model operates in the latent space trained by another model—in our work we use the latent space of JT-VAE. The model can generate molecules with desired properties, as shown on the example of structural and physicochemical properties. The generated molecules are close to the starting ones and the degree of similarity can be controlled via a hyperparameter. In the task of constrained optimization of drug-like molecules our model significantly outperforms previous results. In the future work we plan to extend the approach to multi-parameter optimization of molecules using StarGAN [41]. It would also be interesting to test the model on cases where a small structural change leads to a drastic change in the property (e.g. the so-called activity cliffs) which are hard to model.

## Abbreviations

CADD: computer-aided drug design; VAE: variational autoencoder; GAN: Generative Adversarial Networks; RL: Reinforcement Learning; JT-VAE: Junction Tree Variational Autoencoder; GCPN: Graph Convolutional Policy Network.

## Acknowledgements

We would like to thank Sabina Podlewska for her helpful comments and for fruitful discussions.

## Authors' contributions

LM and AP derived the concept. LM wrote relevant code and performed the experimental work and analysis. AP and JK wrote the paper. KR provided feedback and critical input. TD prepared the revised version of the manuscript. MW did a critical revision of the manuscript. All authors read and approved the final manuscript.

## Funding

Not applicable.

## Availability of data and materials

All source code and datasets used to produce the reported results can be found online at: <https://github.com/ardigen/mol-cycle-gan>.

## Competing interests

The authors declare that they have no competing interests.

## Author details

<sup>1</sup> Ardigen, Podole 76, 30-394 Cracow, Poland. <sup>2</sup> Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Cracow, Poland.

Received: 22 August 2019 Accepted: 16 December 2019

Published online: 08 January 2020

## References

- Ratti E, Trist D (2001) The continuing evolution of the drug discovery process in the pharmaceutical industry. *Farmaco* 56(1–2):13–19. [https://doi.org/10.1016/S0014-827X\(01\)01019-9](https://doi.org/10.1016/S0014-827X(01)01019-9)
- Rao VS, Srinivas K (2011) Modern drug discovery process: an in silico approach. *J Bioinform Seq Anal* 2(5):89–94
- Bajorath J (2002) Integration of virtual and high-throughput screening. *Nat Rev Drug Discov* 1(11):882–894. <https://doi.org/10.1038/nrd941>
- Lavecchia A, Di Giovanni C (2013) Virtual screening strategies in drug discovery: a critical review. *Curr Med Chem* 20(23):2839–2860

5. Honório KM, Moda TL, Andricopulo AD (2013) Pharmacokinetic properties and *in silico* adme modeling in drug discovery. *J Med Chem* 9(2):163–176
6. de Ruyck J, Brysbaert G, Blossey R, Lensink MF (2016) Molecular docking as a popular tool in drug design, an *in silico* travel. *Adv Appl Bioinform* 9:1–11
7. Segler MH, Preuss M, Waller MP (2018) Planning chemical syntheses with deep neural networks and symbolic ai. *Nature* 555(7698):604
8. Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T (2018) The rise of deep learning in drug discovery. *Drug Discov Today*. <https://doi.org/10.1016/j.drudis.2018.01.039>
9. Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, Adams RP (2015) Convolutional networks on graphs for learning molecular fingerprints. In: *Advand neurology*, pp. 2224–2232
10. Jastrzębski S, Leśniak D, Czarnecki WM (2016) Learning to smile (s). *arXiv preprint arXiv:1602.06289*
11. Coley CW, Barzilay R, Green WH, Jaakkola TS, Jensen KF (2017) Convolutional embedding of attributed molecular graphs for physical property prediction. *J Chem Inf Model* 57(8):1757–1772
12. Pham T, Tran T, Venkatesh S (2018) Graph memory networks for molecular activity prediction. *arXiv preprint arXiv:1801.02622*
13. Segler MH, Kogej T, Tyrchan C, Waller MP (2017) Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent Sci* 4(1):120–131. <https://doi.org/10.1021/acscentsci.7b00512>
14. Bjerrum EJ, Threlfall R (2017) Molecular generation with recurrent neural networks (rnns). *arXiv preprint arXiv:1705.04612*
15. Winter R, Montanari F, Noé F, Clevert D-A (2019) Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem Sci*. <https://doi.org/10.1039/C8SC04175J>
16. 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. <https://doi.org/10.1002/minf.201700111>
17. Arús-Pous J, Blaschke T, Ulander S, Reymond J-L, Chen H, Engkvist O (2019) Exploring the gdb-13 chemical space using deep generative models. *J Cheminform* 11(1):20
18. Popova M, Isayev O, Tropsha A (2018) Deep reinforcement learning for *de novo* drug design. *Sci Adv* 4(7):7885
19. Kusner MJ, Paige B, Hernández-Lobato JM (2017) Grammar variational autoencoder. In: *Proceedings of the 34th international conference on machine learning*, volume 70, pp. 1945–1954
20. Dai H, Tian Y, Dai B, Skiena S, Song L (2018) Syntax-directed variational autoencoder for structured data. *arXiv preprint arXiv:1802.08786*
21. Arús-Pous J, Johansson S, Prykhodko O, Bjerrum EJ, Tyrchan C, Reymond J-L, Chen H, Engkvist O (2019) Randomized SMILES strings improve the quality of molecular generative models. *ChemRxiv*
22. Olivecrona M, Blaschke T, Engkvist O, Chen H (2017) Molecular *de-novo* design through deep reinforcement learning. *J Cheminform* 9(1):48
23. Li Y, Vinyals O, Dyer C, Pascanu R, Battaglia P (2018) Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*
24. Li Y, Zhang L, Liu Z (2018) Multi-objective *de novo* drug design with conditional graph generative model. *J Cheminform* 10(1):33
25. Lim J, Hwang S-Y, Kim S, Moon S, Kim WY (2019) Scaffold-based molecular design using graph generative model. *arXiv preprint arXiv:1905.13639*
26. Kingma DP, Welling M (2013) Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*
27. 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 Cent Sci* 4(2):268–276
28. Samanta B, Abir D, Jana G, Chattaraj PK, Ganguly N, Rodriguez MG (2019) Nevae: a deep generative model for molecular graphs. *Proc AAAI Conf Artif Intell* 33:1110–1117
29. Simonovsky M, Komodakis N (2018) Graphvae: towards generation of small graphs using variational autoencoders. *arXiv preprint arXiv:1802.03480*
30. Jin W, Barzilay R, Jaakkola T (2018) Junction tree variational autoencoder for molecular graph generation. In: *Dy, J., Krause, A. (eds.) Proceedings of the 35th international conference on machine learning. Proceedings of machine learning research*, vol. 80. PMLR, Stockholm, Sweden, pp. 2323–2332
31. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: *Advance neurology*, pp. 2672–2680
32. 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*
33. Sanchez-Lengeling B, Outeiral C, Guimaraes GL, Aspuru-Guzik A (2017) Optimizing distributions over molecular space. In: *An objective-reinforced generative adversarial network for inverse-design chemistry (organic)*
34. De Cao N, Kipf T (2018) Molgan: an implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*
35. You J, Liu B, Ying Z, Pande V, Leskovec J (2018) Graph convolutional policy network for goal-directed molecular graph generation. In: *Advances in neural information processing systems*, pp. 6410–6421
36. Prykhodko O, Johansson S, Kotsias P-C, Arús-Pous J, Bjerrum EJ, Engkvist O, Chen H (2019) A *de novo* molecular generation method using latent vector based generative adversarial network
37. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232
38. Maziarka Ł, Pocha A, Kaczmarczyk J, Rataj K, Warchoń M (2019) Mol-cyclegan—a generative model for molecular optimization. In: *Tetko IV, Kůrková V, Karpov P, Theis F (eds) Artificial neural networks and machine learning—ICANN 2019: Workshop and Special Sessions*. Springer, Cham, pp 810–816
39. Weininger D (1988) Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comp Sci* 28(1):31–36
40. Mao X, Li Q, Xie H, Lau RY, Wang Z, Paul Smolley S (2017) Least squares generative adversarial networks. In: *2017 IEEE international conference on computer vision (ICCV)*, pp. 2794–2802 <https://doi.org/10.1109/ICCV.2017.304>
41. Choi Y, Choi M, Kim M, Ha J-W, Kim S, Choo J (2017) Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv:1711.09020*
42. Perarnau G, van de Weijer J, Raducanu B, Álvarez JM (2016) Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*
43. Sterling T, Irwin JJ (2015) Zinc 15—ligand discovery for everyone. *J Chem Inf Model* 55(11):2324–2337. <https://doi.org/10.1021/acs.jcim.5b00559>
44. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. *J Chem Inf Model* 50(5):742–754. <https://doi.org/10.1021/ci100050t>
45. Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Mutowo P, Atkinson F, Bellis LJ, Cibrián-Uhalte E, Davies M, Dedman N, Karlsson A, Magariños MP, Overington JP, Papadatos G, Smit I, Leach AR (2016) The ChEMBL database in 2017. *Nucleic Acids Res* 45(D1):945–954. <https://doi.org/10.1093/nar/gkw1074>; <http://oup.prod.sis.lan/nar/article-pdf/45/D1/D945/8846762/gkw1074.pdf>
46. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*
47. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd international conference on international conference on machine learning*, volume 37. *ICML'15*, pp. 448–456. <http://dl.acm.org/citation.cfm?id=3045118.3045167>
48. Besnard J, Ruda GF, Setola V, Abecassis K, Rodriguez RM, Huang X-P, Norval S, Sassano MF, Shin AI, Webster LA (2012) Automated design of ligands to polypharmacological profiles. *Nature* 492(7428):215. <https://doi.org/10.1038/nature11691>
49. Bickerton GR, Paolini GV, Besnard J, Muresan S, Hopkins AL (2012) Quantifying the chemical beauty of drugs. *Nat Chem* 4(2):90
50. Landrum G (2016) Rdkit: Open-source cheminformatics software

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.