


 Cite this: *RSC Adv.*, 2023, **13**, 36048

# Mol-Zero-GAN: zero-shot adaptation of molecular generative adversarial network for specific protein targets†

 Ravipas Aphikulvanich,<sup>a</sup> Natapol Pornputtpong<sup>b</sup> and Duangdao Wichadakul<sup>\*ac</sup>

Drug discovery is a process that finds new potential drug candidates for curing diseases and is also vital to improving the wellness of people. Enhancing deep learning approaches, e.g., molecular generation models, increases the drug discovery process's efficiency. However, there is a problem in this field in creating drug candidates with desired properties such as the quantitative estimate of druglikeness (QED), synthetic accessibility (SA), and binding affinity (BA), and there is a challenge for training a generative model for specific protein targets that has less pharmaceutical data. In this research, we present Mol-Zero-GAN, a framework that aims to solve the problem based on Bayesian optimization (BO) to find the model optimal weights' singular values, factorized by singular value decomposition, and generate drug candidates with desired properties with no additional data. The proposed framework can produce drugs with the desired properties on protein targets of interest by optimizing the model's weights. Our framework outperforms the state-of-the-art methods sharing the same objectives. Mol-Zero-GAN is publicly available at <https://github.com/cucpbioinfo/Mol-Zero-GAN>.

Received 13th June 2023

Accepted 27th November 2023

DOI: 10.1039/d3ra03954d

[rsc.li/rsc-advances](https://rsc.li/rsc-advances)

## 1 Introduction

The COVID-19 pandemic has shown how vital drug discovery is by discovering novel or repurposing existing drugs for treating diseases and illnesses. Drug discovery not only has the potential to handle emerging diseases but also provides the possibility of finding better replacements for existing ones in terms of efficacy and non-toxicity. It offers humankind many benefits by improving people's health.

With recent research results, machine learning approaches have shown effectiveness in drug discovery. By taking the discovery as a computational problem, drug molecules can be represented as SMILES strings, molecular graphs, or molecular fingerprints. Several studies applied machine learning techniques to learn and produce various molecular representations computationally. For example, recurrent neural network (RNN), an architecture for understanding sequential data patterns, was applied to build text generation models that generate SMILES strings.<sup>1,2</sup> A variational autoencoder (VAE) is another model that learns the probability distribution of training data for sampling

new instances and has been successfully applied to various molecular representations.<sup>3–8</sup> Others<sup>3,9–11</sup> demonstrated the power of a generative adversarial network (GAN), an architecture composed of two neural networks, a generator, and a discriminator, that are trained by competing with each other for generating new samples which are similar to training data but never seen before. Hence, deep learning has proven vital in moving drug discovery forward. Despite the efficiency of the deep learning models, generating drugs with desired properties, such as QED, SA, and BA for a protein target of interest, is still a challenge in this field, and finding new drug candidates for a protein target with fewer drug data is also a big problem. There are three practical approaches to resolving the issues. First, latent space optimization uses trained models that create continuous space representation, then utilizes it with effective querying methods to obtain generated molecules with specific properties such as Bayesian optimization,<sup>7,12–15</sup> swarm optimization.<sup>16</sup> Additionally, some work applied methods of searching for a promising molecule set to train during the training process for optimizing drug generative model such as Blanchard *et al.*,<sup>17</sup> OptiMol,<sup>18</sup> which proposed a fascinating workflow that generates novel molecules with desired properties well. OptiMol applied Conditioning by Adaptive Sampling (CbAS) to shift the based learned distribution to maximize an objective function: the docking score of the protein target. DrugGEN Ünlü *et al.*<sup>19</sup> can generate molecules with specific properties by feeding protein features into its graph transformer decoder module. The provided protein features allow DrugGEN to design target-specific molecules by incorporating the desired protein target information. In addition to this approach, Brookes

<sup>a</sup>Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, 10330, Thailand. E-mail: ravipas.apk@gmail.com

<sup>b</sup>Department of Biochemistry and Microbiology, Faculty of Pharmaceutical Sciences, Chulalongkorn University, Bangkok, 10330, Thailand. E-mail: natapol.p@chula.ac.th

<sup>c</sup>Center of Excellence in Systems Biology, Faculty of Medicine, Chulalongkorn University, Bangkok, 10330, Thailand. E-mail: duangdao.w@chula.ac.th

† Electronic supplementary information (ESI) available. See DOI: <https://doi.org/10.1039/d3ra03954d>



*et al.*<sup>20</sup> proposed a model-based adaptive sampling method to optimize the generation of molecules with desired properties.<sup>20</sup> Second, reinforcement learning (RL) is a powerful approach for decision-making in a molecular generation, where the current molecular structure serves as the state, the decision on which structure to add is the action, and the molecular property serves as the reward.<sup>21,22</sup> Third, fine-tuning the Generative Pretrained Transformer (GPT), a versatile model capable of various tasks, has garnered significant attention recently.<sup>23–25</sup> While its primary applications have been in natural language processing, there's growing interest in adapting GPT for other domains through fine-tuning.<sup>26,27</sup> One such promising adaptation is PETrans, a method that fine-tunes GPT for *de novo* drug design.<sup>28</sup> PETrans leverages transfer learning and protein-specific encoding to generate target-specific ligands, offering a fresh perspective on the challenge of producing drug-like molecules optimized for binding to target proteins.

However, optimizing a single objective may be inefficient as other features may conflict after the optimization. For example, some studies indicated a result of single-objective optimization on binding affinity compared to multi-objective optimization. Goel *et al.* proposed MoleGuLAR, a framework that applied RL to optimize an RNN model to generate desired drug candidates. MoleGuLAR presented alternating rewards, satisfying multi-objective optimization by swapping an objective during optimization. Its results showed that the single objective optimization improved the focused feature but did not perform well on other features such as druglikeness and octanol–water partition coefficient (LogP). On the other hand, multi-objective optimization approaches covered more of different properties. Therefore, proper optimization is essential for enhancing the models to generate drugs with desired properties.

Instead of adjusting the latent space directly or generating additional compounds to refine the training process or using an RL, in this paper, we present a framework, Mol-Zero-GAN (Fig. 1), inspired by FSGAN,<sup>29</sup> which learned from a few images to adapt the singular values of the pre-trained weights in StyleGAN2 (ref. 30) for synthesizing images in the targeted domain while keeping the natural style of the base model. Mol-Zero-GAN aims to optimize a drug-generative model's weights to enhance the model's ability to generate new drug candidates with desired properties without additional data while preserving the quality of other properties. We combined the modifying idea with Bayesian optimization to find optimal singular weight values in LatentGAN<sup>31</sup> to obtain a drug

generative model specially optimized for required properties. We have mentioned our significant contributions as follows:

- We propose a new method for optimizing the pre-trained drug generative model, which has improved compared to the base pre-trained model in terms of molecular properties.
- The proposed method outperformed or achieved on-par performance compared with the related works.
- The proposed method requires no additional data for generating new drugs targeting a specific protein of interest while preserving the druglikeness and synthetic accessibility of the base pre-trained model.

## 2 Methodology

### 2.1 Base molecular generator

We used the pre-trained generator of LatentGAN as a molecular generator in our proposed framework. LatentGAN is a drug generation model that integrates GAN and an autoencoder by utilizing the pre-trained autoencoder as a molecular representation and training GAN to learn to generate latent vectors, which are then decoded by the pre-trained decoder of the autoencoder. In this work, we trained the pre-trained LatentGAN by following the MOSES benchmark instructions,<sup>32</sup> which consists of a dataset and a set of hyperparameters, and then we optimized LatentGAN's generator in our proposed framework to satisfy our objectives.

### 2.2 Mol-Zero-GAN

We propose Mol-Zero-GAN (Fig. 2), a framework for optimizing a pre-trained drug generative model to generate drug candidates for specific protein targets. We applied singular value decomposition (SVD) on weights of the pre-trained generator's five layers to extract the singular values, which we will use as parameters for Bayesian optimization (BO) to find the best replacement of these values of the model's layers to satisfy the defined objective function, formulated as eqn (1):

$$x^* = \operatorname{argmax}_{x \in X} f(x) \quad (1)$$

where  $f$  is the objective function,  $x$  is a replacement for singular values in the pre-trained LatentGAN's generator,  $x^*$  is the optimal parameters, and  $X$  is the domain of  $x$  determined by eqn (2):

$$X = \{M \cdot b | b \in h\} \quad (2)$$

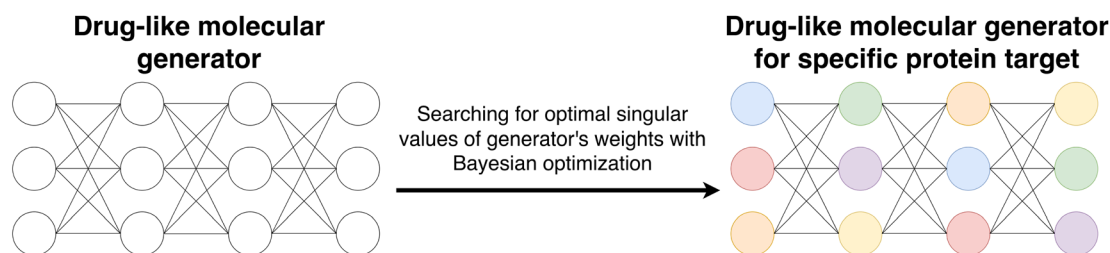


Fig. 1 Overview of Mol-Zero-GAN's framework.



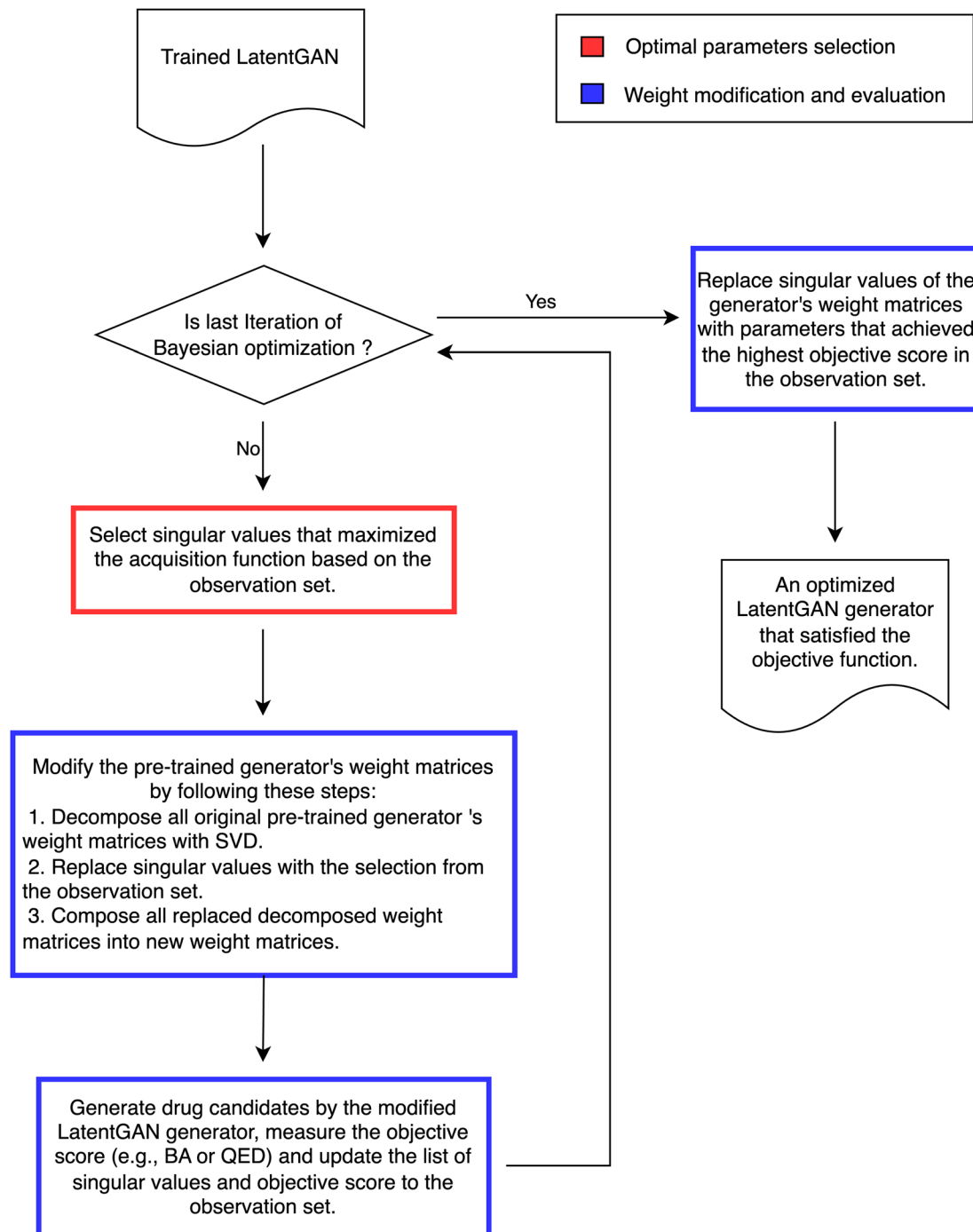


Fig. 2 Mol-Zero-GAN overall workflow.

where  $M$  is the initial first  $k$  singular values per generator weight layers, and  $h$  is the range of multipliers. The following subsections describe how we optimized and evaluated the singular values of the model's layers in details.

**2.2.1 Singular values optimization and search space reduction.** We applied BO to find optimal singular values to replace their base values to obtain the optimized model (Algorithm 1). We used the Successive Response Surface Method (SRSM)<sup>33</sup> to iteratively reduce the search space of  $x$  as shown in

Algorithm 2, where  $x_t$  represents the current optimal parameters, and  $\gamma_{\text{pan}}$ ,  $\gamma_{\text{osc}}$ , and  $\eta$  are reduction parameters,  $x_i$  denotes the best parameters at iteration  $i$ , and  $x_0$  represents the initial parameters. In addition, we adopted the Upper Confidence Bound (UCB)<sup>34</sup> as an acquisition function, which balances the exploitation of the optimal points with the exploration of unexplored regions of the search space (eqn (3)).

$$\text{UCB}(x) = \mu_V(x) + \kappa\sigma_V(x) \quad (3)$$



where  $\mu_v(x)$  and  $\sigma_v(x)$  are the estimated mean and standard deviation values of taking  $x$  as parameters, based on past observations  $V$ , and  $\kappa$  is a constant that controls the balance between the exploration and exploitation by applying the weight or importance of standard deviation value in the process of decision-making.

---

**Algorithm 1:** Mol-Zero-GAN's Bayesian optimization
 

---

**Input:** Objective function  $f$ , molecular generation model  $G$ , iteration  $t$   
 Initialize observation set  $V \leftarrow \emptyset$   
 Initialize parameters  $x_0$  with original singular values of  $G$   
**for**  $i = 0$  to  $t$  **do**  
   Evaluate parameters with objective function:  
    $y = f(x_i)$   
   Add observation point to observation list:  
    $V \cup (x_i, y)$   
   Select next parameters  $x_{i+1}$  based on  $V$  with BO and UCB  
   Reduce parameters domain with SRSB by updating lower and upper bounds  
**return**  $\text{argmax}_x f(x)$

---



---

**Algorithm 2:** Domain reduction at BO iteration with SRSB
 

---

**Input:** Current optimal parameters  $x_t$ , reduction parameters  $\gamma_{pan}$ ,  $\gamma_{osc}$ , and  $\eta$   
**Given:**  $x_i$  is the best parameters at iteration  $t$ ,  $x_0$  is the initial parameters, and  $[lowerbound_0, upperbound_0]$  is the range of initial bound of the Bayesian optimization.  
 Compute the change in distance between the current and previous optimal points:  
 $d_t = 2 \cdot (x_t - x_{t-1}) / r_{t-1}$   
 Compute the search space contraction rate:  
 $c = d_t \cdot d_{t-1}$   
 $\hat{c} = \sqrt{c} \cdot \text{sign}(c)$   
 $\gamma = 0.5 \cdot (\gamma_{pan} \cdot (1.0 + \hat{c}) + \gamma_{osc} \cdot (1.0 - \hat{c}))$   
 Update the range of parameters:  
 $r_t = (\eta + d_t \cdot (\gamma - \eta)) \cdot r_{t-1}$   
 Compute the lower and upper bound of the new range of parameters:  
 $lowerbound_t = \max(x_t - r_t/2, lowerbound_{t-1})$   
 $upperbound_t = \min(x_t + r_t/2, upperbound_{t-1})$   
**return**  $[lowerbound_t, upperbound_t]$

---

**2.2.2 Singular values evaluation and replacement.** To modify the generator's weight and evaluate the score, we applied SVD on every layer of the model to access the singular values, then replaced them with a replacement list to get an optimized model, and evaluated the objective function score by measuring the average molecular properties' scores of the generated samples from the optimized model (Algorithm 3).

In this work, we experimented with three objective functions, e.g., QED, BA, and weighted sum, on each selected protein target. We defined the objective function as the respective single

properties for QED and BA scores. For weighted sum, we aggregated QED and BA into a new objective function by calculating the weighted sum of each property's score formulated as eqn (4):

$$f(x) = -\frac{\alpha(\overline{\text{BA}} - b)}{m - b} + \beta(\overline{\text{QED}}) \quad (4)$$

where  $f$  is the objective function,  $x$  is parameters,  $\overline{\text{BA}}$  and  $\overline{\text{QED}}$  are the average BA and QED scores of the generated compounds,  $m$  is the average BA score from the BA optimization experiment, and  $b$  is the average BA score of the base model,  $\alpha$ ,  $\beta$  are constants that we determined by balancing the difference in ranges between the expected range of  $-\frac{(\overline{\text{BA}} - b)}{m - b}$  and  $\overline{\text{QED}}$ . The expected range for weighted sum optimization of  $-\frac{(\overline{\text{BA}} - b)}{m - b}$  is typically 0 to 1 because we expected the lowest value of  $\overline{\text{BA}}$  to be equal to the base model and the highest value of  $\overline{\text{BA}}$  to be equal to the current BA of the optimized model. Additionally, based on the mean and standard deviation of the base model, we expected the range for  $\overline{\text{QED}}$  to be between 0.60 and 1. However, since we aimed to maintain the QED value of generated compounds, we narrowed the expected range to 0.60 to 0.80. To balance the weight of both BA and QED, we chose  $\alpha = 0.2$  and  $\beta = 1$ , from the difference in the range of expected values of  $-\frac{(\overline{\text{BA}} - b)}{m - b}$  and  $\overline{\text{QED}}$ , which is 1 (range of 0 to 1) and 0.2 (range of 0.60 to 0.80), respectively.

---

**Algorithm 3:** Objective Function Evaluation
 

---

**Input:** Replacement of singular value  $x$ , molecular properties evaluation  $\mathbf{q}$ , molecular generator  $\mathbf{G}$ , number of replacements per layer  $\mathbf{k}$ , number of samples  $\mathbf{n}$   
**for**  $i = 0$  to  $\text{len}(\text{layer of } G)$  **do**  
   Assign  $A$  as a layer  $i^{\text{th}}$  of  $G$   
   Perform SVD on  $A$ :  $A = U\Sigma V^T$   
   Replace first  $k$  of  $A$  values with  $x$ 's slice array value from  $x[k \times i, k \times (i + 1)]$ :  $\Sigma^* \leftarrow \Sigma$   
   Compose new weight matrix with new singular matrix:  $A^* = U\Sigma^* V^T$   
   Update  $A^*$  to layer  $i^{\text{th}}$  of  $G$ :  $G \leftarrow A^*$   
 Sample  $n$  molecules with the updated  $G$   
**return**  $q(n \text{ molecules})/n$

---

### 2.3 Molecular metrics

This work considers four molecular properties for the evaluation: BA, QED, SA, and diversity. The BA value can refer to the strength of interaction between a drug candidate and a protein target; a more negative value indicates a stronger interaction. A QED value in a range of 0 to 1 represents druglikeness. The QED closer to 1, the more drug-like. SA is a value ranging from 1 to 10, the higher value the more difficulty synthesizing the drug. Finally, diversity is a metric that reflects how compounds are diverse within the generated set.



Table 1 Grid parameters for docking calculations

Protein PDB ID	Grid size	Grid box center (x, y, z)
DRD3	36.975 Å × 39.940 Å × 38.529 Å	(10.192, 22.963, 23.884)
SARS-CoV-2 M <sup>Pro</sup>	40 Å × 40 Å × 40 Å	(−10.063, 16.667, 67.294)
TTBK1	40 Å × 40 Å × 40 Å	(61.955, 18.155, 24.305)
EGFR	44 Å × 49 Å × 57 Å	(19.496, 35.001, 89.270)
S1PR1	75 Å × 75 Å × 75 Å	(120.713, 118.886, 131.755)
HTR1A	75 Å × 75 Å × 75 Å	(93.496, 92.635, 76.821)
AKT1	18 Å × 18 Å × 18 Å	(−22.454, 3.192, 9.571)

Table 2 Average QED score on different number of singular values in optimization process

Number of singular values	Average QED score
1	0.82
2	0.87
3	0.86
4	0.87
5	<b>0.88</b>
6	0.87
7	0.86
8	0.84
9	0.87
10	0.83

## 2.4 Data preparation and experimental setup

As our objectives were to generate new drugs having lower BA for seven protein targets to compare with the previous works, *i.e.*, OptiMol, MoleGuLAR, PETrans, and DrugGEN, while preserving other properties, we obtained the 3D structures of the target proteins DRD3 (PDB: 3PBL) from the Directory of Useful Decoys, Enhanced (DUD-E),<sup>35</sup> TTBK1 (PDB: 4BTK), SARS-CoV-2 M<sup>Pro</sup> (PDB: 6LU7), EGFR (PDB: 2RGP), HTR1A (PDB: 7E2X), S1PR1 (PDB: 7VIH) from the Protein Data Bank Center<sup>36</sup> and AKT1 (PDB: 4GV1) from a web server for structure-based kinase profiling called ProfKin.<sup>37</sup> We created a grid parameter for AKT1 by manually drawing a grid box around the center point, corresponding to the binding site's center of mass. Additionally, we adopted the drug molecule's docking grid of other proteins, as shown in Table 1 from OptiMol, MoleGuLAR, and PETrans. Based on the prepared protein structures, for each iteration of the model's singular values optimization and

evaluation, we converted SMILES strings generated from the optimized model to 3D structures using the RDKit library<sup>38</sup> and prepare\_ligand4.py from ref. 39. We then used AutoDock Vina<sup>40</sup> to determine the binding affinity (BA) with the most stable configuration of molecular poses for proteins DRD3, SARS-CoV-2 M<sup>Pro</sup>, TTBK1, and AKT1, and used QuickVina-W<sup>41</sup> for proteins EGFR, HTR1A, and S1PR1. For QED and SA, we applied RDKit and MOSES to measure the scores of validly generated compounds. Finally, we measured the diversity by calculating the average pairwise Tanimoto distance among the generated compounds' fingerprints.

In this study, we evaluated our results by measuring the molecular properties: BA, QED, SA, and diversity of validly generated samples, 100 000 compounds for DRD3, 10 000 compounds each for TTBK1, SARS-CoV-2 M<sup>Pro</sup>, EGFR, HTR1A, S1PR1 and AKT1 to compare with previous works. We used Pytorch<sup>42</sup> as the development tool for training and manipulating the generator's weights, and bayes\_opt,<sup>43</sup> a Python library for performing Bayesian optimization.

## 3 Results

In this section, we present the results of our experiments on QED, BA, and weighted sum optimization for protein targets: DRD3, TTBK1, SARS-CoV-2 M<sup>Pro</sup>, EGFR, HTR1A, S1PR1, and AKT1. After that, we discuss and compare our results with the base model before optimization and previous works.

### 3.1 Hyperparameters for model optimization

Based on our computational resources, we selected the number of singular values per layer ( $k$ ) by pivoting to perform our method with  $n = 3072$  and QED as the objective function. First, we tested values of  $k$  from 1 to 10 (Table 2) and selected the

Table 3 Hyperparameters

Hyperparameter	Value
Number of singular values per layer for replacement ( $k$ )	5
Number of generated drug molecules for calculating the objective score ( $n$ )	3072
Number of iterations in the BO	50
UCB exploration parameter ( $\kappa$ )	2.576
SRSM oscillation parameter ( $\gamma_{osc}$ )	0.7
SRSM panning parameter ( $\gamma_{pan}$ )	1.0
SRSM scaling parameter ( $\eta$ )	0.9
Range of multipliers ( $h$ )	[0, 2]



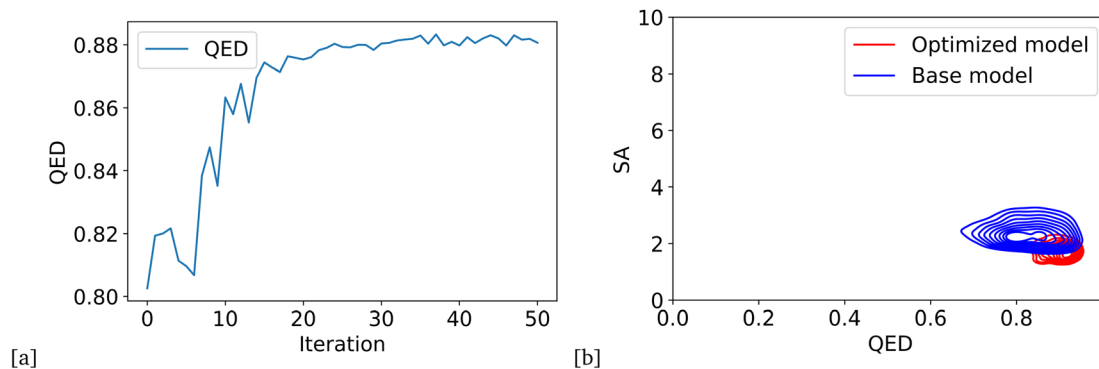


Fig. 3 QED optimization results (a) improvement of average QED score over optimization iteration (b) the distribution of average QED and SA of validly generated compounds compared between the base and the optimized model.

value,  $k = 5$ , that gave us the best average QED score. Next, we chose the number of generated drug molecules ( $n$ ) to fit our available computational resources to ensure we could run the process until completion and used the default SRSM's hyperparameters of bayes\_opt. Finally, we decided the multiplier range ( $h$ ) to be  $[0, 2]$  with the lower bound as zero since singular values are nonnegative and the upper bound as two because it is a symmetric scaling factor. Table 3 shows the hyperparameters we used in our Bayesian optimization process.

### 3.2 Optimized model with QED objective

By defining the average QED of generated compounds as an objective function, we obtained an optimized model that can produce drug candidates with higher druglikeness than the base model. Fig. 3a presents the improvement of average QED

through iterations of Bayesian optimization, and Fig. 3b shows the distribution of the QED and SA of valid compounds generated by the optimized model compared to the base model. The results show that the QED optimization has successfully improved both the QED and SA of generated drug candidates from the optimized model compared to the base model.

### 3.3 Optimized model with BA objective

We optimized the base model to generate drug candidates with lower binding affinities for all protein targets: DRD3, SARS-CoV-2 M<sup>Pro</sup>, TTBK1, EGFR, HTR1A, S1PR1, and AKT1 by defining the average BA score of the generated compounds as an objective function. Fig. 4 and 5 show the changes in BA and QED scores between consecutive BO iterations, BA distribution, and QED and SA distribution of valid compounds generated by the base

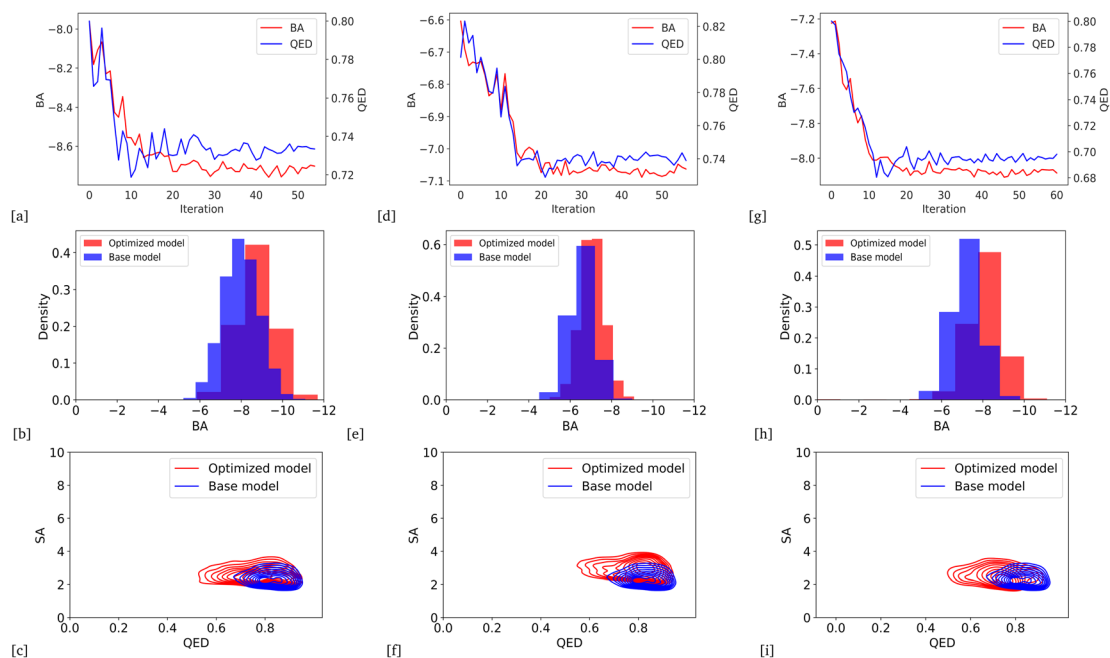


Fig. 4 Change of BA and QED scores between consecutive BO iterations, BA distribution, and QED and SA distribution of generated compounds by DRD3-optimized model compared to the base model (a)–(c), SARS-CoV-2 M<sup>Pro</sup> (d)–(f), and TTBK1 (g)–(i) from BA optimization results.



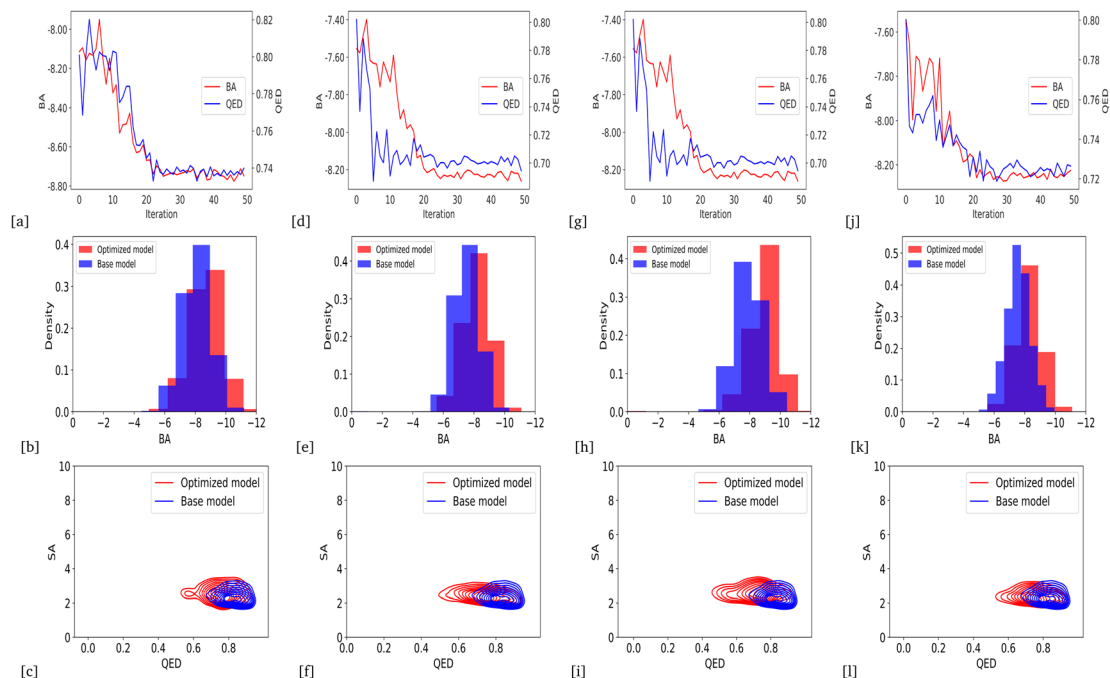


Fig. 5 Change of BA and QED scores between consecutive BO iterations, BA distribution, and QED and SA distribution of generated compounds by EGFR-optimized model compared to the base model (a)–(c), HTR1A (d)–(f), and S1PR1 (g)–(i), AKT1 (j)–(l) from BA optimization results.

model compared to the DRD3-optimized model (Fig. 4a–c), the SARS-CoV-2 M<sup>PRO</sup>-optimized model (Fig. 4d–f), the TTBK1-optimized model (Fig. 4g–i), the EGFR-optimized model (Fig. 5a–c), the HTR1A-optimized model (Fig. 5d–f), the S1PR1-optimized

model (Fig. 5g–i), and the AKT1-optimized model (Fig. 5j–l). From the results, BA optimization has shown improvement in the BA scores of generated drug candidates from optimized models of the protein targets compared to the base model.

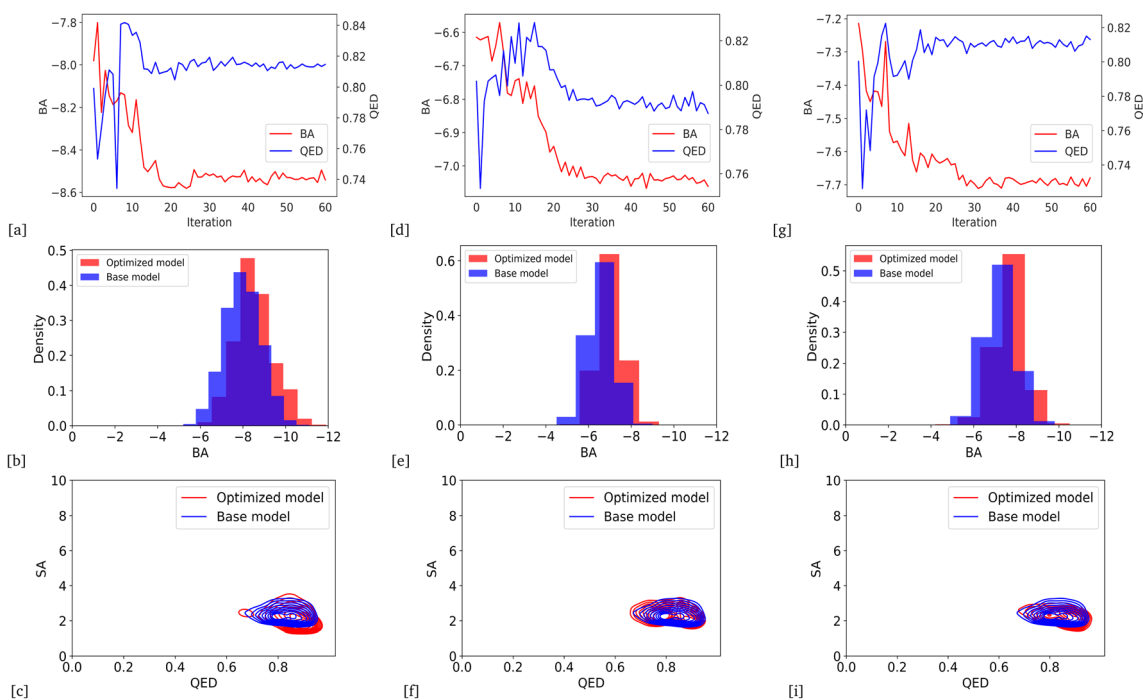


Fig. 6 Change of BA and QED scores between consecutive BO iterations, BA distribution, and QED and SA distribution of generated compounds by DRD3-optimized model compared to the base model (a)–(c), SARS-CoV-2 M<sup>PRO</sup> (d)–(f), and TTBK1 (g)–(i) from weighted sum optimization results.



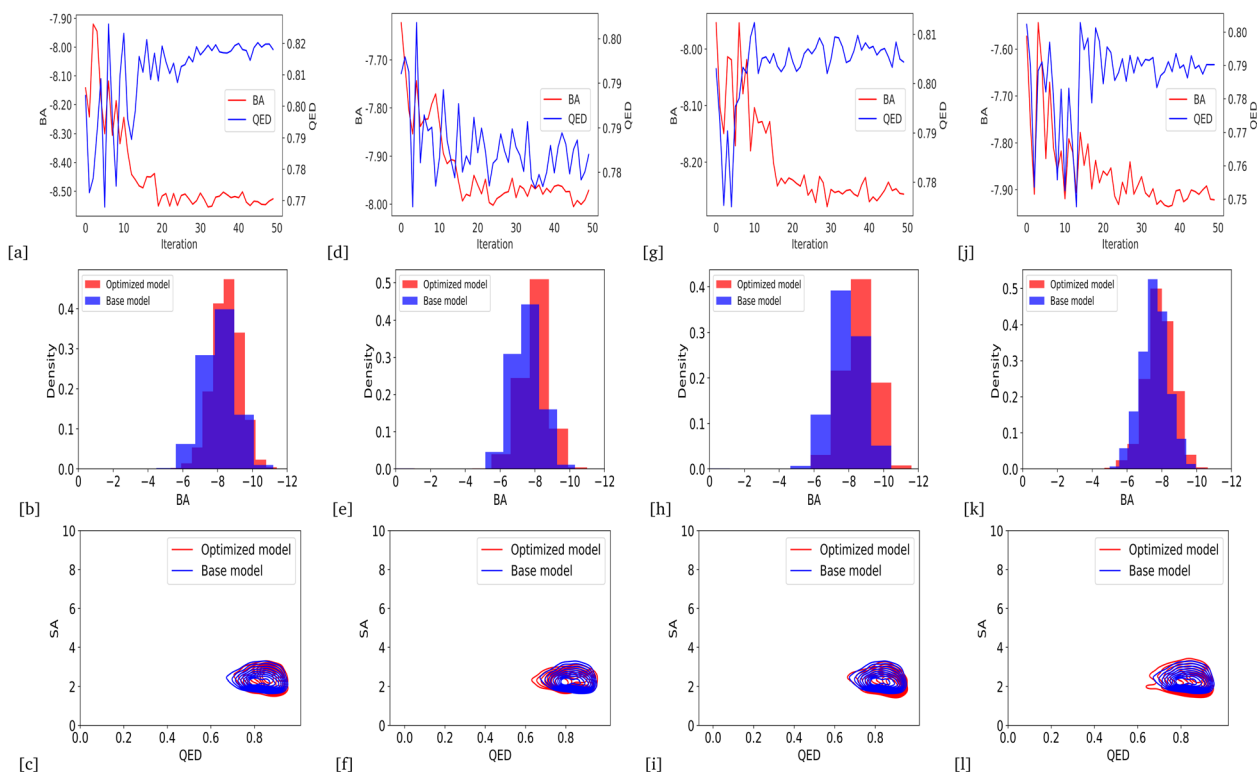


Fig. 7 Change of BA and QED scores between consecutive BO iterations, BA distribution, and QED and SA distribution of generated compounds by EGFR-optimized model compared to the base model (a)–(c), HTR1A (d)–(f), and S1PR1 (g)–(i), AKT1 (j)–(l) from weighted sum optimization results.

However, there is a noticeable variance in other properties of the generated drugs, especially QED. This observation suggests that our model behaves in a way that indicates a trade-off when optimizing for BA.

### 3.4 Optimized model with weighted sum of BA and QED

Due to the higher variance of QED scores in our models based on BA optimization, we integrated average QED and BA scores into a weighted sum objective function (eqn (4)) with  $\alpha = 0.2$  and  $\beta = 1$ . As a result, we could generate the lower average BA score by optimizing the models for all protein targets: DRD3, TTBK1, SARS-CoV-2 M<sup>Pro</sup>, EGFR, HTR1A, S1PR1, and AKT1, with an average QED score comparable to the base model. Fig. 6 and 7 show the changes in BA and QED scores between consecutive BO iterations, BA distribution, and QED and SA distribution of valid compounds generated by the base model compared to the

DRD3-optimized model (6a–c), the SARS-CoV-2 M<sup>Pro</sup>-optimized model (Fig. 6d–f), the TTBK1-optimized model (Fig. 6g–i), the EGFR-optimized model (Fig. 7a–c), the HTR1A-optimized model (Fig. 7d–f), the S1PR1-optimized model (Fig. 7g–i), and the AKT1-optimized model (Fig. 7j–l). The results show successful improvement of BA score for Mol-Zero-GAN-generated compounds while preserving the base model's QED scores for all seven protein targets compared to BA optimization. However, by employing self-feedback mechanisms to fine-tune GANs for generating drugs with enhanced binding affinities, there is a potential for an increased variance in the QED. This increase can be attributed to the fact that the model is adjusted based on the data it generates rather than solely relying on the initial training dataset. This iterative and dynamic adjustment can lead to broader exploration in the generated molecular space.

Table 4 Mean and standard deviation of BA, QED, SA, and diversity scores of 100 000 valid drug candidates generated by Mol-Zero-GAN's optimized model for DRD3 protein target compared to OptiMol

Model	BA(std)	QED(std)	SA(std)	Diversity
Base model (LatentGAN)	−7.96(0.97)	0.80(0.10)	2.52(0.55)	0.86
OptiMol (multiobjective) <sup>a</sup>	−8.58	0.69	2.84	<b>0.86</b>
Mol-Zero-GAN (BA optimization)	− <b>8.72(0.89)</b>	0.73(0.13)	2.77(0.64)	0.83
Mol-Zero-GAN (weighted sum optimization)	−8.53(0.89)	<b>0.81(0.12)</b>	<b>2.36(0.67)</b>	0.82

<sup>a</sup> This information was obtained directly from OptiMol's results.



**Table 5** Mean and standard deviation of BA, QED, SA, and diversity scores of 10 000 valid drug candidates generated by Mol-Zero-GAN's optimized model for SARS-CoV-2 M<sup>Pro</sup> protein target compared to MoleGuLAR

Model	BA(std)	QED(std)	SA(std)	Diversity
Base model (LatentGAN)	-6.61(0.60)	<b>0.80(0.10)</b>	<b>2.51(0.54)</b>	<b>0.86</b>
MoleGuLAR <sup>a</sup>	-7.80( <b>0.94</b> )	0.11(0.13)	4.25(0.98)	0.81
Mol-Zero-GAN (BA optimization)	-7.07(0.58)	0.74(0.14)	2.99(0.58)	0.84
Mol-Zero-GAN (weighted sum optimization)	-7.04(0.57)	0.79(0.10)	2.54(0.63)	0.84

<sup>a</sup> We reproduced the results of MoleGuLAR by cloning the source code from its official repository at GitHub. Then, we applied a multi-objective optimization approach with alternating rewards. We ran the optimization process for 175 iterations by modifying the original approach and setting BA and QED with a target of 1 as the combined objectives instead of BA and LogP with a target of 2.5. We swapped every 35 iterations, starting with BA. Like the original work, we used AutoDock-GPU<sup>44</sup> docking score for BA evaluation during the reinforcement learning process and used AutoDock Vina to evaluate the generated compounds.

**Table 6** Mean and standard deviation of BA, QED, SA, and diversity scores of 10 000 valid drug candidates generated by Mol-Zero-GAN's optimized model for TTBK1 protein target compared to MoleGuLAR

Model	BA(std)	QED(std)	SA(std)	Diversity
Base model (LatentGAN)	-7.20(0.72)	0.80(0.10)	2.51(0.54)	<b>0.86</b>
MoleGuLAR <sup>a</sup>	-9.37( <b>1.08</b> )	0.13(0.10)	3.33(0.78)	0.80
Mol-Zero-GAN (BA optimization)	-8.07(0.88)	0.70(0.14)	2.76(0.98)	0.83
Mol-Zero-GAN (weighted sum optimization)	-7.69(0.69)	<b>0.81(0.10)</b>	<b>2.44(0.60)</b>	0.83

<sup>a</sup> We reproduced the results of MoleGuLAR by cloning the source code from its official repository at GitHub. Then, we applied a multi-objective optimization approach with alternating rewards. We ran the optimization process for 175 iterations by modifying the original approach and setting BA and QED with a target of 1 as the combined objectives instead of BA and LogP with a target of 2.5. We swapped every 35 iterations, starting with BA. Like the original work, we used AutoDock-GPU<sup>44</sup> docking score for BA evaluation during the reinforcement learning process and used AutoDock Vina to evaluate the generated compounds.

**Table 7** Mean and standard deviation of BA, QED, SA, and diversity scores of 10 000 valid drug candidates generated by Mol-Zero-GAN's optimized model for EGFR protein target compared to PErans

Model	BA(std)	QED(std)	SA(std)	Diversity
Base model (LatentGAN)	-8.07(0.92)	0.80(0.10)	2.51(0.54)	<b>0.86</b>
PErans <sup>a</sup>	-7.97	0.45	2.74	N/A
Mol-Zero-GAN (BA optimization)	-8.70( <b>1.05</b> )	0.74(0.12)	2.71(0.69)	0.84
Mol-Zero-GAN (weighted sum optimization)	-8.52(0.79)	<b>0.82(0.10)</b>	<b>2.47(0.53)</b>	0.83

### 3.5 Benchmark results with previous works

Tables 4–10 show the molecular properties' statistical summary of our work compared to the base model and previous works. Both BA and weight sum optimization outperformed the previous works regarding QED and SA. Additionally, the BA optimization surpassed PErans (S1PR1) and OptiMol (DRD3) in terms of BA performance. ESI Fig. S1–S7<sup>†</sup> show examples of generated molecules of BA-optimized models for DRD3, SARS-CoV-2 M<sup>Pro</sup>, TTBK1, EGFR, HTR1A, S1PR1, AKT1 and ESI Fig. S8–S14<sup>†</sup> show examples of generated molecules of weighted sum optimization models of DRD3, SARS-CoV-2 M<sup>Pro</sup>, TTBK1, EGFR, HTR1A, S1PR1, and AKT1 respectively.

## 4 Discussions

Bayesian optimization (BO) is an optimization process that performs well in low-dimensional search space problems. For our work, BO is suitable for searching the optimal singular

values, including 25 parameters (five layers with five singular values per layer). The singular value modification process modifies the singular values of the diagonal matrices of the generator's weights while keeping the left and right matrices the same, preserving the main characteristics of each layer's linear transformation. Hence, Mol-Zero-GAN can potentially optimize target properties while maintaining others.

From the QED optimization result, Mol-Zero-GAN has successfully leveraged the average QED value compared to the base model. By setting QED as an objective function, the optimization process motivated the optimized model to generate drug candidates with a better average QED score. Higher QED indicates better drug-like properties that boost the optimized model to create drug candidates with properties similar to real-world existing drugs. Furthermore, since the properties of existing drugs refer to the potential for large-scale production, the optimized model is also implicitly motivated to improve the average SA score indicating the more straightforward synthesis of the drug candidates.



**Table 8** Mean and standard deviation of BA, QED, SA, and diversity scores of 10 000 valid drug candidates generated by Mol-Zero-GAN's optimized model for HTR1A protein target compared to PETrans

Model	BA(std)	QED(std)	SA(std)	Diversity
Base model (LatentGAN)	-7.50(0.83)	<b>0.80(0.10)</b>	2.51(0.54)	<b>0.86</b>
PETrans <sup>a</sup>	<b>-8.59</b>	0.53	2.97	N/A
Mol-Zero-GAN (BA optimization)	-8.16(0.91)	0.70(0.14)	2.60(0.50)	0.84
Mol-Zero-GAN (weighted sum optimization)	-7.99(0.83)	0.78(0.11)	<b>2.45(0.51)</b>	0.84

<sup>a</sup> This information was obtained directly from PETrans results.

**Table 9** Mean and standard deviation of BA, QED, SA, and diversity scores of 10 000 valid drug candidates generated by Mol-Zero-GAN's optimized model for S1PR1 protein target compared to PETrans

Model	BA(std)	QED(std)	SA(std)	Diversity
Base model (LatentGAN)	-7.90(0.92)	0.80(0.10)	2.51(0.54)	<b>0.86</b>
PETrans <sup>a</sup>	<b>-9.58</b>	0.46	2.56	N/A
Mol-Zero-GAN (BA optimization)	-8.94(1.03)	0.68(0.14)	2.83(1.02)	0.82
Mol-Zero-GAN (weighted sum optimization)	-8.20(0.85)	<b>0.81(0.11)</b>	<b>2.41(0.58)</b>	0.84

<sup>a</sup> This information was obtained directly from PETrans results.

Mol-Zero-GAN also successfully optimized the BA score in the BA optimization process. As a result, the optimized model could generate drug candidates with an average BA score lower than the base model. However, the average QED score of the optimized model was not as good as the base model's and fluctuated more, reflecting the optimization trade-off between BA and QED.

The introduced weighted sum optimization can generate drug candidates with an average lower BA score than the base model but not as good as the BA optimization. However, the average QED is similar to the base model, reflecting the optimization trade-off between BA and QED and the effectiveness of modeling the objective function to motivate the optimized model to preserve QED. The introduced weighted sum optimization also maintained SA from the base model, as shown in the

generated drug properties, which have similar values to the base model. The  $\alpha$  and  $\beta$  are balancing factors determining the trade-off between BA optimization and QED preservation, which are adjustable to specific proposes.

Our work has outperformed previous works in terms of drug-likeness and synthesis-accessible properties. By comparing our work to the multi-objective model from OptiMol, our optimized model from BA optimization generated drug candidates with better properties in terms of average BA, QED, and SA. Although the generated drugs from Mol-Zero-GAN's optimized models did not get average BA scores as low as the MoleGuLAR, our optimized models generated drugs with more practical QED and SA properties. OptiMol and MoleGuLAR rely on retraining a generative model and reinforcement learning. Retraining a generative model by selecting generated molecules with lower

**Table 10** Mean and standard deviation of BA, QED, SA, and diversity scores of 10 000 valid drug candidates generated by Mol-Zero-GAN's optimized model for AKT1 protein target compared to DrugGEN

Model	BA(std)	QED(std)	SA(std)	Diversity
Base model (LatentGAN)	-7.57(0.79)	<b>0.80(0.10)</b>	2.51(0.54)	0.86
DrugGEN-Prot <sup>a</sup>	<b>-8.54(1.23)</b>	0.53(0.19)	3.67(0.89)	<b>0.88</b>
DrugGEN-CrossLoss <sup>a</sup>	-8.24(1.15)	0.54(0.20)	3.28(0.93)	<b>0.88</b>
DrugGEN-Ligand <sup>a</sup>	-8.32(1.28)	0.51(0.18)	3.28(0.97)	<b>0.88</b>
DrugGEN-RL <sup>a</sup>	-8.45(1.20)	0.53(0.20)	3.10(1.03)	0.87
DrugGEN-NoTarget <sup>a</sup>	-8.10(1.17)	0.57(0.21)	3.30(1.04)	<b>0.88</b>
Mol-Zero-GAN (BA optimization)	-8.25(0.81)	0.72(0.12)	2.57(0.56)	0.84
Mol-Zero-GAN (weighted sum optimization)	-7.91(0.80)	0.79(0.11)	<b>2.50(0.63)</b>	0.83

<sup>a</sup> The DrugGEN models and data were sourced from the DrugGEN's GitHub repository (<https://github.com/HUBioDataLab/DrugGEN>). DrugGEN employs two primary Generative Adversarial Networks (GANs) of multilayer perceptron (MLP) and graph transformers: GAN1 is responsible for crafting new molecular structures, and GAN2 refines these molecules to target specific properties. The framework is diversified into five distinct models: DrugGEN-Prot merges GAN1's molecule generation with GAN2's protein design capabilities; DrugGEN-CrossLoss solely utilizes GAN1 to align generated molecules with real inhibitors; DrugGEN-Ligand, a variant of DrugGEN-Prot, integrates AKT1 inhibitor features into GAN2; DrugGEN-RL, an advanced version of DrugGEN-Ligand, promotes molecular diversity through reinforcement learning; and finally, DrugGEN-NoTarget employs GAN1 to replicate molecules from the ChEMBL database without any specific targeting.



BA and higher QED to retrain the model in OptiMol would modify the model weight without trying to preserve the original model's characteristics that learned various properties from the actual drug dataset during the initial training. On the other hand, reinforcement learning in MoleGuLAR tries to learn a policy gradient that takes the generator into a model space, which generates new molecules that maximize the reward scores. This optimized generator, however, may fall into the model space with an outstanding property but violating other properties. Even though MoleGuLAR's alternative rewards try to switch back and forth between properties' rewards, updating the model space for the first property's rewarding may constrain modifying the model space of the second property's rewards.

Compared to DrugGEN, the model is specifically designed to focus on small molecules and incorporates protein features like AKT1. However, including such protein features might alter certain characteristics inherent to the DrugGEN's base model during drug design. In comparison, Mol-Zero-GAN keeps its base model's features stable and consistent, offering an advantage over DrugGEN regarding drug-likeness. While PETrans modifies large models like GPT with a small dataset for drug design, Mol-Zero-GAN stands out because it doesn't need extra data for fine-tuning. Instead, it naturally preserves the base model's features, ensuring consistent drug-likeness without relying on additional datasets.

The comparison results of Mol-Zero-GAN with OptiMol, MoleGuLAR, DrugGEN, and PETrans reflect Mol-Zero-GAN's ability to preserve the base model characteristics, making it generate drug candidates with comparable or better BA and better QED and SA compared to the previous works.

## 5 Conclusion

This research presents Mol-Zero-GAN, a framework for optimizing a drug generation model to generate drugs with desired properties based on Bayesian optimization of model weight matrices' singular values. We demonstrate its ability to minimize the binding affinity while preserving other objectives of the base model without additional training data. Compared to the recent research results with similar purposes, all Mol-Zero-GAN's optimized models, based on a single or weighted sum objective, produced comparable or outperformed previous results. For future work, we plan to elaborate on the objective functions and reduce the time for BA evaluation.

## Conflicts of interest

There are no conflicts to declare.

## Acknowledgements

This work was fully supported by the Program Management Unit of Competitiveness (PMUC), Office of National Higher Education Science Research and Innovation Policy Council, Thailand, under contract number C10F640366. We want to thank Chulalongkorn University Technology Center (UTC) and

Mahidol AI Center for facilitating the GPU infrastructure for our model development and testing.

## References

- 1 M. H. Segler, T. Kogej, C. Tyrchan and M. P. Waller, *ACS Cent. Sci.*, 2018, **4**, 120–131.
- 2 R. van Deursen, P. Ertl, I. V. Tetko and G. Godin, *J. Cheminf.*, 2020, **12**, 1–14.
- 3 M. Skalic, D. Sabbadin, B. Sattarov, S. Sciabola and G. De Fabritiis, *Mol. Pharmaceutics*, 2019, **16**, 4282–4291.
- 4 B. Samanta, A. De, G. Jana, V. Gómez, P. K. Chattaraj, N. Ganguly and M. Gomez-Rodriguez, *Journal of Machine Learning Research*, 2020, **21**, 4556–4588.
- 5 J. Lim, S. Ryu, J. W. Kim and W. Y. Kim, *J. Cheminf.*, 2018, **10**, 1–9.
- 6 Q. Liu, M. Allamanis, M. Brockschmidt and A. Gaunt, *Advances in neural information processing systems*, 2018, vol. 31.
- 7 M. J. Kusner, B. Paige and J. M. Hernández-Lobato, *International conference on machine learning*, 2017, pp. 1945–1954.
- 8 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2018, **4**, 268–276.
- 9 A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper and A. Zhavoronkov, *Mol. Pharmaceutics*, 2017, **14**, 3098–3104.
- 10 G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias and A. Aspuru-Guzik, *arXiv*, 2017, preprint arXiv:1705.10843, DOI: [10.48550/arXiv.1705.10843](https://doi.org/10.48550/arXiv.1705.10843).
- 11 B. Sanchez-Lengeling, C. Outeiral, G. Guimaraes and A. Aspuru-Guzik, *ChemRxiv*, 2017, preprint, DOI: [10.26434/chemrxiv.5309668.v3](https://doi.org/10.26434/chemrxiv.5309668.v3).
- 12 G.-B. R. W. J. Duvenaud and D. H.-L. J. Sánchez-Lengeling, Automatic chemical design using a data-driven continuous representation of molecules, *ACS Cent. Sci.*, 2018, **4**, 268.
- 13 W. Jin, R. Barzilay and T. Jaakkola, *International conference on machine learning*, 2018, pp. 2323–2332.
- 14 H. Dai, Y. Tian, B. Dai, S. Skiena and L. Song, *arXiv*, 2018, preprint, arXiv:1802.08786, DOI: [10.48550/arXiv.1802.08786](https://doi.org/10.48550/arXiv.1802.08786).
- 15 K. Korovina, S. Xu, K. Kandasamy, W. Neiswanger, B. Poczos, J. Schneider and E. Xing, *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 3393–3403.
- 16 R. Winter, F. Montanari, A. Steffen, H. Briem, F. Noé and D.-A. Clevert, *Chem. Sci.*, 2019, **10**, 8016–8024.
- 17 A. E. Blanchard, C. Stanley and D. Bhowmik, *J. Cheminf.*, 2021, **13**, 1–8.
- 18 J. Boitreaud, V. Mallet, C. Oliver and J. Waldispühl, *J. Chem. Inf. Model.*, 2020, **60**, 5658–5666.
- 19 A. Ünlü, E. Çevrim, A. Sarıgün, H. Çelikbilek, H. A. Güvenilir, A. Koyaş, D. C. Kahraman, A. Rifaioğlu and A. Olğaç, *arXiv*, 2023, preprint, arXiv:2302.07868, DOI: [10.48550/arXiv.2302.07868](https://doi.org/10.48550/arXiv.2302.07868).
- 20 D. Brookes, H. Park and J. Listgarten, *International conference on machine learning*, 2019, pp. 773–782.



- 21 T. Blaschke, J. Arús-Pous, H. Chen, C. Margreitter, C. Tyrchan, O. Engkvist, K. Papadopoulos and A. Patronov, *J. Chem. Inf. Model.*, 2020, **60**, 5918–5922.
- 22 M. Goel, S. Raghunathan, S. Laghuvarapu and U. D. Priyakumar, *J. Chem. Inf. Model.*, 2021, **61**, 5815–5826.
- 23 B. Banar and S. Colton, *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, 2022, pp. 19–35.
- 24 OpenAI, *GPT-4 Technical Report*, 2023.
- 25 S. Wu, O. Irsoy, S. Lu, V. Dabrovolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg and G. Mann, *BloombergGPT: A Large Language Model for Finance*, 2023.
- 26 J.-S. Lee and J. Hsiang, *World Pat. Inf.*, 2020, **62**, 101983.
- 27 N. Alexandr, O. Irina, K. Tatyana, K. Inessa and P. Arina, *Data Science and Intelligent Systems: Proceedings of 5th Computational Methods in Systems and Software 2021*, 2021, vol. 2, pp. 748–757.
- 28 X. Wang, C. Gao, P. Han, X. Li, W. Chen, A. Rodríguez Patón, S. Wang and P. Zheng, *Int. J. Mol. Sci.*, 2023, **24**, 1146.
- 29 E. Robb, W.-S. Chu, A. Kumar and J.-B. Huang, *arXiv*, 2020, preprint, arXiv:2010.11943, DOI: [10.48550/arXiv.2010.11943](https://doi.org/10.48550/arXiv.2010.11943).
- 30 T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.
- 31 O. Prykhodko, S. V. Johansson, P.-C. Kotsias, J. Arús-Pous, E. J. Bjerrum, O. Engkvist and H. Chen, *J. Cheminf.*, 2019, **11**, 1–13.
- 32 D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, *et al.*, *Front. Pharmacol.*, 2020, **11**, 565644.
- 33 N. Stander and K. Craig, *Engineering Computations*, 2002, **19**, 431–450.
- 34 T. L. Lai, H. Robbins, *et al.*, *Adv. Appl. Math.*, 1985, **6**, 4–22.
- 35 M. M. Mysinger, M. Carchia, J. J. Irwin and B. K. Shoichet, *J. Med. Chem.*, 2012, **55**, 6582–6594.
- 36 H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne, *Nucleic Acids Res.*, 2000, **28**, 235–242.
- 37 Z. Shen, Y.-H. Yan, S. Yang, S. Zhu, Y. Yuan, Z. Qiu, H. Jia, R. Wang, G.-B. Li and H. Li, *Eur. J. Med. Chem.*, 2021, **225**, 113772.
- 38 G. Landrum, *et al.*, *RDKit: a software suite for cheminformatics, computational chemistry, and predictive modeling*, 2013.
- 39 G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell and A. J. Olson, *J. Comput. Chem.*, 2009, **30**, 2785–2791.
- 40 O. Trott and A. J. Olson, *J. Comput. Chem.*, 2010, **31**, 455–461.
- 41 N. M. Hassan, A. A. Alhossary, Y. Mu and C.-K. Kwok, *Sci. Rep.*, 2017, **7**, 15451.
- 42 A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.
- 43 F. Nogueira, *et al.*, *Bayesian Optimization: open source constrained global optimization tool for Python*, 2014, accessed: 5 Nov 2023.
- 44 D. Santos-Martins, L. Solis-Vasquez, A. F. Tillack, M. F. Sanner, A. Koch and S. Forli, *J. Chem. Theory Comput.*, 2021, **17**, 1060–1073.

