A Model Details

A.1 Example of Diversified SMILES

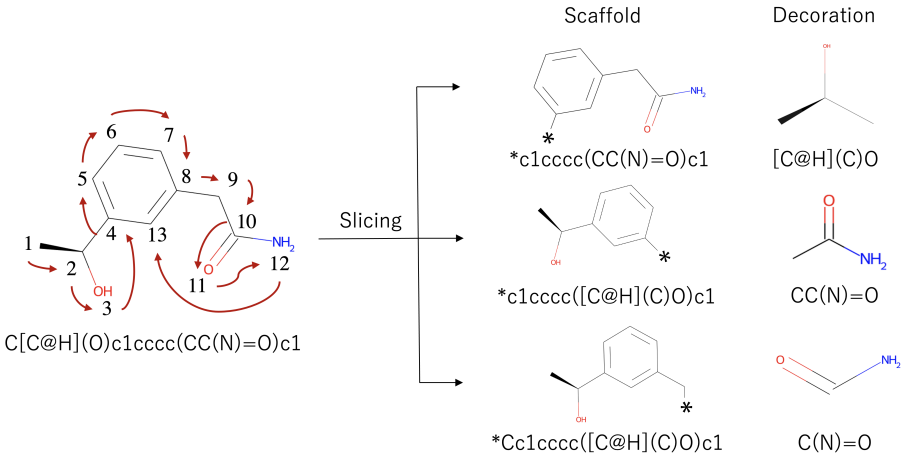


Fig. A.1. An example of diversified SMILES.

Figure A.1 illustrates three ways of slicing a SMILES string. In this case, the canonical SMILES string is "C[C@H](O)c1cccc(CC(N)=O)c1." The numbers and arrows on the canonical molecular structure in the left panel indicate the traversal order of the atoms. The right panel shows the scaffolds and decorations after slicing the molecular structure between atoms 2 and 4, 8 and 9, and 9 and 10. Decorator is fully trained on different slice representations of the same molecule, which resolves the problem of substantial variation in SMILES strings.

A.2 Self-attention Mechanism

The atomic positions are input to the following sinusoidal positional encoding function:

$$\begin{aligned} \mathrm{PE}_{(pos,2i)} &= \sin(pos/10000^{2i}/d_{model}), \\ \mathrm{PE}_{(pos,2i+1)} &= \cos(pos/10000^{2i}/d_{model}), \end{aligned} \tag{8}$$

where pos is the unique position of each atom and i is the i-th dimension of the atomic embedding dimension d_{model}. Finally, the positional encoding is added to the bottom of the decoder stack along with the scaffold and decoration embeddings.

The transformer decoder consists of a stack of decoder layers. Each decoder layer consists of two multi-head attention modules with corresponding layer normalization and a feedforward neural network (see middle panel of Fig. 1). The

scaffold providing the global scaffold features is input to the multi-head attention (② in Fig. 1). The masked multi-head attention module masks the atoms beyond position i to prevent future information exposure. Its input is the decoration (③ in Fig. 1) that captures the local features between the generated atoms and the global semantics of each generated atom and the scaffold.

Concretely, the input representations of the scaffold and decoration consist of query, key, and value matrics, denoted as $\boldsymbol{Q} \in$, $\boldsymbol{K}$, and $\boldsymbol{V}$. The self-attention is given by

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V}, \tag{9}$$

where d_k represents the dimension of the key matrix. The output of the multi-head attention module is then fed into a feedforward neural network to generate the decorations. Finally, the generated decorations and scaffolds are combined into SMILES representations of molecules (④ in Fig. 1).

A.3 Overview Algorithm

Algorithm 1 shows an overview of the proposed models. First, the generator G_θ is pre-trained by maximum likelihood estimation on the real SMILES dataset $\mathcal{D}_r$. Next, the generator produces dataset $\mathcal{D}_z$ with the same number of SMILES strings as $\mathcal{D}_r$ to balance the two datasets. The two datasets are shuffled before pre-training the discriminator D_ϕ. Finally, the generator and the discriminator are alternately trained in adversarial training and parameters θ are updated with an MC search using the policy gradient.

B Experiment Details

B.1 Diversity Calculation

The diversity was calculated from the Tanimoto similarity between the Morgan fingerprints of any two novel molecules in the generated set. Let V_i and V_j be the Morgan fingerprints of two arbitrary generated novel molecules. The Tanimoto similarity is then defined as

$$\text{Sim}(V_i, V_j) = \frac{|V_i \& V_j|}{|V_i| + |V_j| - |V_i \& V_j|},$$

where $|\cdot|$ represents the number of bits set in the fingerprints, and & is the common bits in the two fingerprints. The diversity is then calculated as

$$\text{Div}(\mathcal{D}_z) = 1 - \frac{1}{|\mathcal{D}_z|} \sum_{V_i, V_j \in \mathcal{D}_z} \text{Sim}(V_i, V_j).$$

Algorithm 1: Procedures of SpotGAN and SpotWGAN

Data: a SMILES dataset $\mathcal{D}_r$
Initialization : the generator G_θ, the discriminator D_ϕ
`// Pre-process the dataset` $\mathcal{D}_r$`.`
Use diversified SMILES to pair scaffold and decoration slices.
Compute segment IDs and offset IDs for the scaffolds and decorations (② and ③ in Fig. 1). `// Pre-train the generator on` $\mathcal{D}_r$`.`
for $i = 1 \rightarrow f\_epochs$ **do**
  Update θ with maximum likelihood estimation (④ in Fig. 1).
end
Generate a dataset $\mathcal{D}_z$.
Shuffle datasets $\mathcal{D}_r$ and $\mathcal{D}_z$ (⑤ in Fig. 1). `// Pre-train the discriminator on the shuffled dataset.`
for $i = 1 \rightarrow d\_epochs$ **do**
  Update ϕ with cross entropy (SpotGAN) or Wasserstein distance (SpotWGAN) by Eq.(7) (⑥ in Fig. 1).
end
`// Adversarial training of` G_θ `and` D_ϕ`.`
for $i = 1 \rightarrow epochs$ **do**
  `// Train the generator` G_θ`.`
  for $j = 1 \rightarrow f\_steps$ **do**
    Generate a dataset $\mathcal{D}_z$ based on the given scaffolds.
    Shuffle the datasets $\mathcal{D}_r$ and $\mathcal{D}_z$.
    `// Check if SpotGAN is executed.`
    if *SpotGAN* **then**
      Compute R^{G_θ} by Eqs. (2) and (5) (⑦, ⑧ in Fig.1).
    end
    `// Check if SpotWGAN is executed.`
    if *SpotWGAN* **then**
      Execute SpotWGAN.
    end
    Update θ by Eq. (6) (⑨ in Fig. 1).
  end
  `// Train discriminator` D_ϕ`.`
  for $k = 1 \rightarrow d\_steps$ **do**
    Update discriminator's parameters of ϕ.
  end
end

B.2 Optimized Properties

Drug-likeness. When calculating the QED scores of molecules, we generally assign different weights to eight molecular descriptors: the molecular weight (MW), octanol-water partition coefficient (ALOGP), number of hydrogen bond donors (HBDs), number of hydrogen bond acceptors (HBAs), molecular polar surface area (PSA), number of rotatable bonds (ROTBs), number of aromatic rings (AROMs), and number of structural alerts (ALERTS). The calculation is as follows:

$$\mathrm{QED} = \exp\left(\frac{\sum_{i=1}^{8} W_i \ln d_i)}{\sum_{i=1}^{8} W_i}\right),$$

where d_i and W_i represent the desirability function and weight of the i-th descriptor, respectively. Usually, the weights of the eight molecular descriptors are obtained through chemical experiments. In practice, the QED score is calculated by a function in the RDKit tool. The larger the QED score, the more drug-like the molecule.

Solubility. In the physical sciences, solubility is quantified by logP, where P is the partition coefficient (defined as the ratio of concentrations of a molecule in a mixture of two immiscible solvents at equilibrium). The logP is calculated as

$$\mathrm{logP} = \log \frac{c_o}{c_w},$$

where c_o and c_w indicate the substance activity in the organic and water phases, respectively. In practice, we calculate the logP of a molecule using the RDKit tool. The larger the logP value, the higher the lipophilicity of the molecule to the organic phase.

Synthesizability. Synthesizability is evaluated in terms of the SA score:

$$\mathrm{SA} = r_s - \sum_{i=1}^{5} p_i,$$

where r_s indicates the "synthetic knowledge" gained by analyzing the features of synthetic molecules. r_s is the ratio of the summed contributions from all fragments to the number of fragments in the molecule. In this work, we calculated r_s from the experimental results [10]. p_i ($i \in \{1, \cdots, 5\}$) represents the ring complexity, stereo complexity, macrocycle penalty, size penalty, and bridge penalty, which were calculated using the RDKit tool. Note that the larger the SA score, the easier is the synthesis of the molecule.

BIO. The BIO score measures the probability of interaction between a molecule and a target protein. In this work, we chose dopamine receptor D2 (DRD2) as the target protein and used a random forest classifier in the Scikit-learn tool [30] to calculate the BIO scores.

Specifically, we first collected DRD2-molecule interaction data from the ChEMBL30 (accessed 2022-3-17) dataset [12]. Then, we used the Ki (unit: nmol/L) values in ChEMBL217 as the interaction data. After excluding missing values and duplicated molecules, we obtained 6652 molecule-Ki data. Next, we built 500 decision-tree classifiers using the random forest classifier for binary classification. We used the SMILES representation of a molecule as input to the classifier, which was then converted to a 2048-dimensional ECFP4 (extended connectivity fingerprint, up to four bonds). The labels of the binary classifier were defined as follows:

$$\text{label} = \begin{cases} 0 & \text{if } \ 9 - \log \text{Ki} > 7, \\ 1 & \text{otherwise.} \end{cases}$$

Finally, the output of the classifier is calculated as the predicted probability with the label of 1, which is the BIO score.

B.3 Drug-likeness Evaluation Results

Figure B.1 shows the QED distributions of the molecular structures on the ZINC dataset. Figures B.2 and B.3 show the 12 molecular structures with the best QED scores on the QM9 and ZINC datasets, respectively.

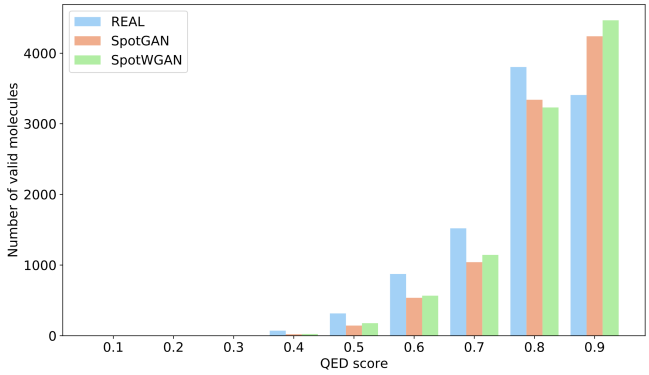
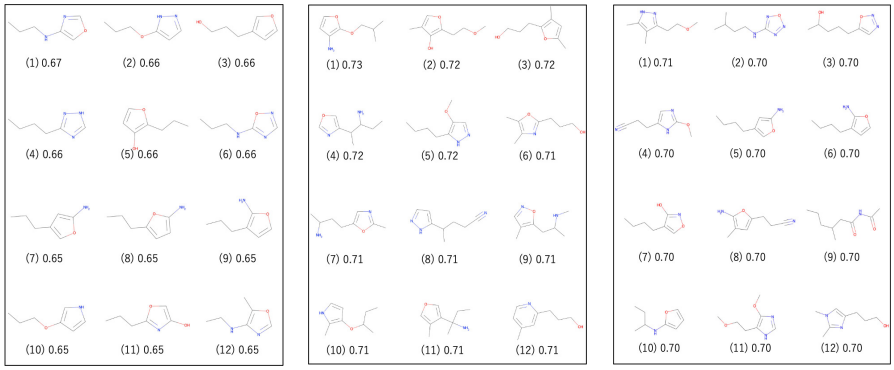


Fig. B.1. QED Distributions with drug-likeness as the optimized property on the ZINC dataset.

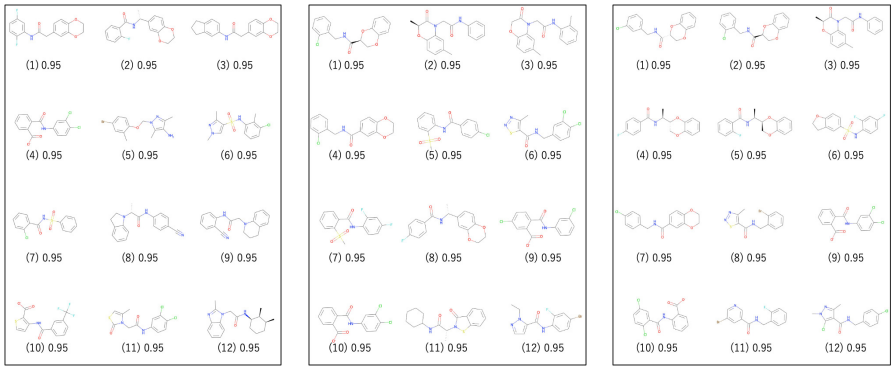
B.4 Solubility Evaluation Results

To verify the usefulness of our models when the optimized property was the solubility, we plotted the logP distributions of our models generated on the QM9 and ZINC datasets. The results are displayed in Fig.e B.4. The validity on the QM9 dataset was below 90%, mainly for the following reasons: (1) The QM9



(a) Training dataset. (b) SpotGAN. (c) SpotWGAN.

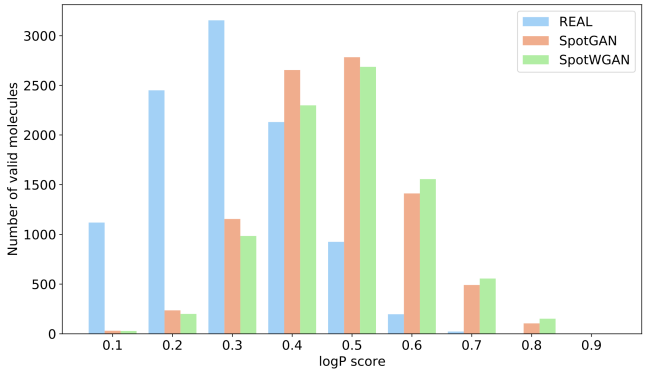
Fig. B.2. Top-12 molecular structures with their QED scores on the QM9 dataset.



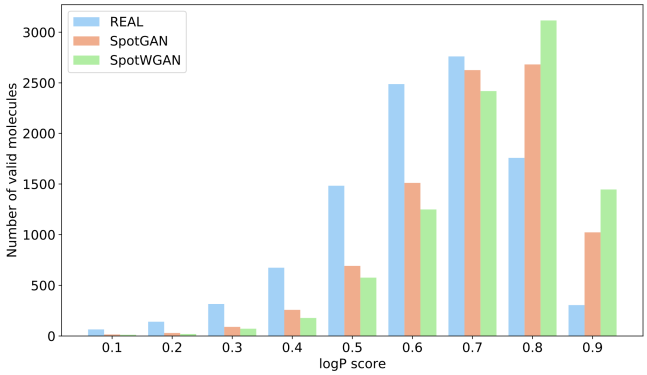
(a) Training dataset. (b) SpotGAN. (c) SpotWGAN.

Fig. B.3. Top-12 molecular structures with their QED scores on the ZINC dataset.

dataset contains many small molecules with limited chemical rules for decorations (average decoration length < 5); (2) the generated molecules (Fig. B.5 and B.6) tended to be lipophilic, resulting in extended carbon chains and ring structures containing more than seven carbon atoms.



(a) Distributions of logP on the QM9 dataset.



(b) Distributions of logP on the ZINC dataset.

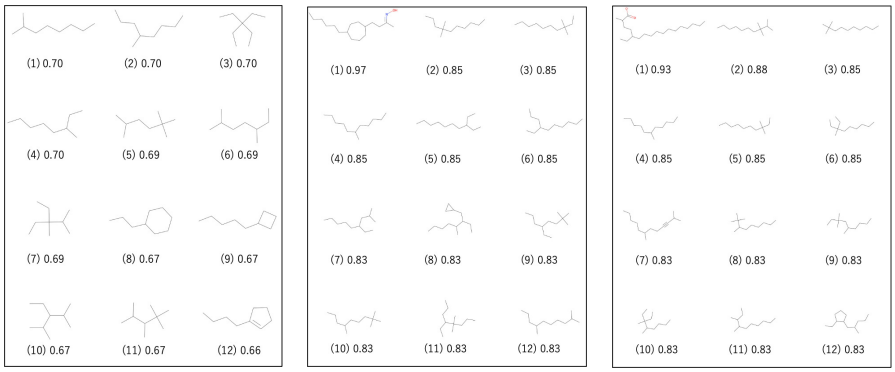
Fig. B.4. Distributions of logP scores with solubility as the optimized property.

B.5 Synthesizability Evaluation Results

Figure B.7 depicts the SA distributions of the molecular structures on both datasets. Figures B.8 and B.9 show the 12 molecular structures with the best SA scores on both datasets, respectively.

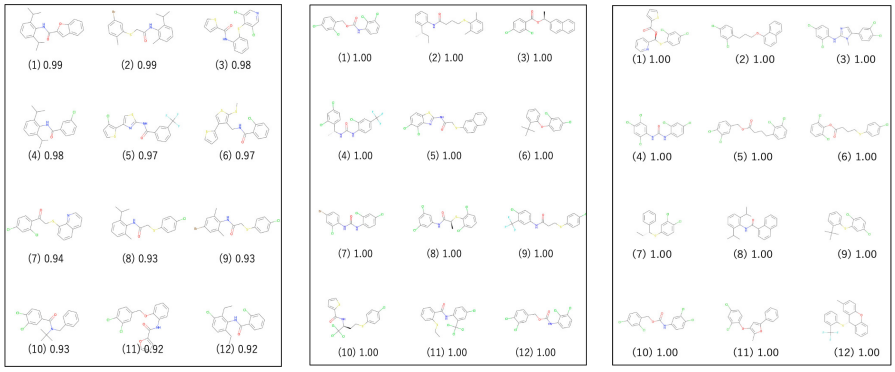
B.6 Property Optimization Curve on ZINC Dataset

Figure B.10 shows the property scores of SpotGAN on the ZINC dataset versus epoch.



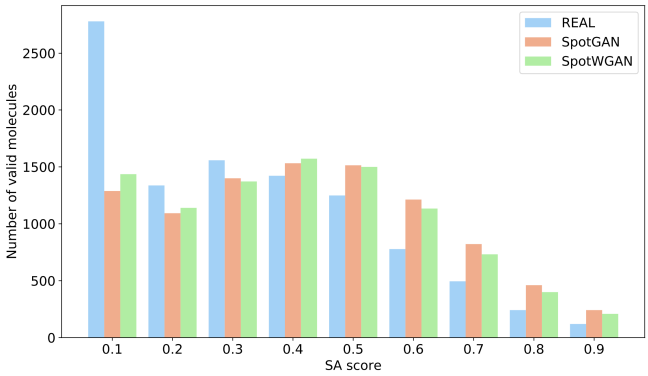
(a) Training dataset. (b) SpotGAN. (c) SpotWGAN.

Fig. B.5. Top-12 molecular structures with their logP scores on the QM9 dataset.

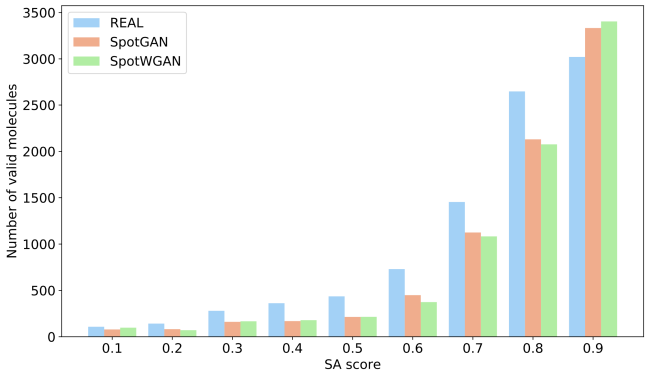


(a) Training dataset. (b) SpotGAN. (c) SpotWGAN.

Fig. B.6. Top-12 molecular structures with their logP scores on the ZINC dataset.

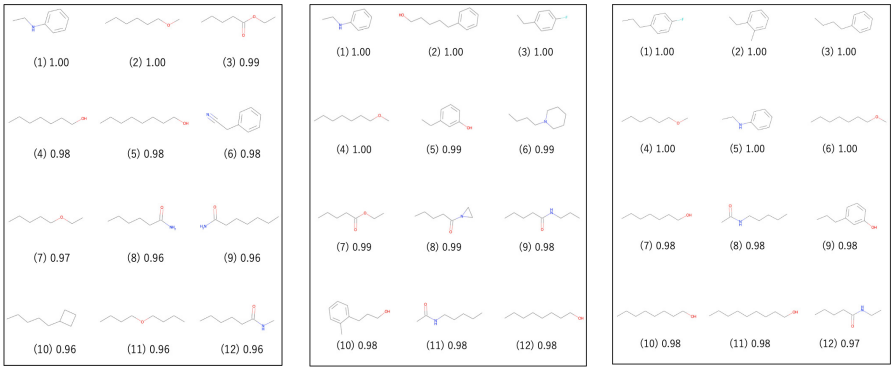


(a) Distributions of SA on the QM9 dataset.



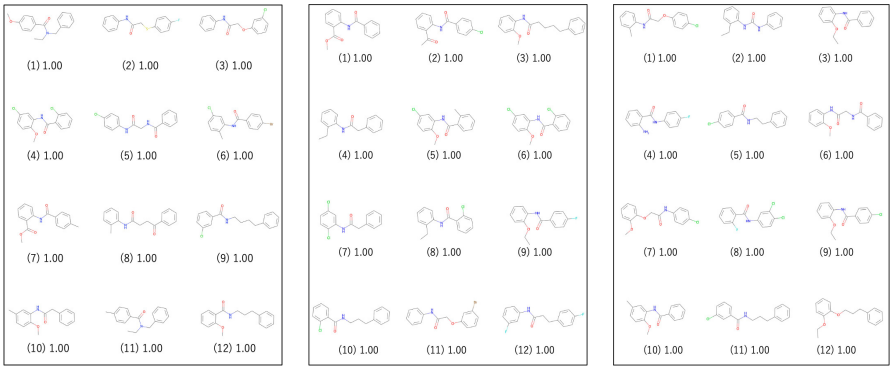
(b) Distributions of SA on the ZINC dataset.

Fig. B.7. Distributions of SA scores with synthesizability as the optimized property.



(a) Training dataset. (b) SpotGAN. (c) SpotWGAN.

Fig. B.8. Top-12 molecular structures with their SA scores on the QM9 dataset.



(a) Training dataset. (b) SpotGAN. (c) SpotWGAN.

Fig. B.9. Top-12 molecular structures with their SA scores on the ZINC dataset.

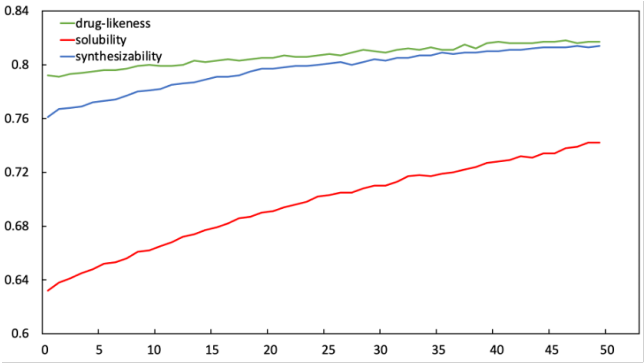


Fig. B.10. Property scores as functions of epoch on the ZINC dataset.

B.7 Case Studies of Bioactivity Optimization

Table B.1. Evaluation results of SpotGAN after optimizing the BIO on the ZINC dataset.

Method	BIO	Validity (%)	Uniqueness (%)	Novelty (%)	Diversity	Time (h)
Decorator	0.22	93.58	97.04	92.42	0.89	1.63
SpotGAN	0.24	94.88	95.51	94.31	0.88	16.90

Table B.1 shows the evaluation results of SpotGAN on the ZINC dataset when the optimized property was BIO. The reinforced training improved the validity, novelty, and BIO scores.