Appendices

A Detailed Model Description

A.1 InstGAN's Design Motivations

We chose an LSTM backbone because it models SMILES strings efficiently [Guo and Schwaller, 2023] while remaining lightweight and stable, advantages that are particularly valuable when molecular data are limited. To verify the soundness of our choice, we benchmarked three Transformer-based alternatives: TransORGAN [Li *et al.*, 2022], SpotGAN [Li and Yamanishi, 2023], and InstGAN* (where LSTM was replaced with Transformer). As summarized in Table A.1, where the best score is shaded and the second-best is under-lined. Our LSTM-based InstGAN outperforms all transformer-based baselines. These results indicate that, for SMILES-based molecular generation under data-constrained settings compared to natural language processing, the additional capacity of the Transformer does not translate into better results, whereas LSTM offers a more efficient solution.

Model	Validity	Uniqueness	Novelty	Total
TransORGAN	74.3	91.8	100.0	68.2
SpotGAN	93.3	92.8	92.8	80.3
InstGAN*	97.8	70.7	98.0	67.8
InstGAN	97.9	98.3	<u>99.7</u>	95.9

Table A.1: Comparison of Transformer- and LSTM-based models.

A.2 Details of Generator

Recurrent neural networks (RNNs) have demonstrated versatility in handling both input and output sequence data. When presented with an input vector at time step t, $\boldsymbol{S}_{1:t} = [s_1, \cdots, s_t]$, and the corresponding output vector $\widetilde{\boldsymbol{S}}_{1:t} = [\widetilde{s}_1, \cdots, \widetilde{s}_t]$, the objective of the RNNs is to model the distribution $p(\widetilde{s}_t | s_1, s_2, \cdots, s_t)$. These networks operate as dynamic systems, as illustrated in Fig. 1 (a), where the RNNs state at any t-th time step (i.e., any t-th position in the SMILES string) relies on the previous observations.

For a given input SMILES string, the generative RNNs undergo training to extend this SMILES string by predicting the subsequent sequence token, denoted as $\widetilde{s}(t) = s_t$. In this study, RNNs equipped with LSTM cells are utilized, a choice made to address challenges like vanishing and exploding gradient problems associated with extended sequences and a substantial network architecture. At each time step t, such a network is delineated by the subsequent set of equations:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{si} s_t + \mathbf{W}_{hi} \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{1}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{sf} s_t + \mathbf{W}_{hf} \mathbf{h}_{t-1} + \mathbf{b}_f), \tag{2}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{so} s_t + \mathbf{W}_{ho} \mathbf{h}_{t-1} + \mathbf{b}_o), \tag{3}$$

$$\widetilde{\mathbf{c}}_t = \tanh(\mathbf{W}_{sc} s_t + \mathbf{W}_{hc} \mathbf{h}_{t-1} + \mathbf{b}_c), \tag{4}$$

$$\mathbf{c}_t = \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \widetilde{\mathbf{c}}_t, \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{c}_t). \tag{6}$$

Here, the vectors $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ denote the input, forget, and output gate vectors, respectively. These vectors are outcomes of the sigmoid activation function, indicating values between 0 and 1. $\mathbf{c}_t$ is a vector responsible for preserving long-term memory. The output vector $\mathbf{y}_t$ is ultimately computed as follows:

$$\widetilde{s}_t = g(\mathbf{W}_{hs} \mathbf{h}_t), \tag{7}$$

where $g(\cdot)$ represents a nonlinear activation function.

The common approach for RNNs in SMILES string generation typically follows a left-to-right (forward) direction, covering the range from $t = 1$ to $t = T$, where T represents the length of a SMILES string. In the training phase, the initial position of the input is filled with a start-of-sequence token, and the final position is filled with an end-of-sequence token. New sequences are generated by (1) inputting the starting token ("<SOS>"), and (2) iteratively selecting the subsequent token as the model progresses through the previous sequence of tokens until the end token ("<EOS>") is produced. At each time step t, the probability for each v-th token to follow the preceding portion of the generated string is computed using a softmax function. The calculation is outlined as follows:

$$p(s_{t+1} = v | s_1, s_2, \cdots, s_t) = \frac{\exp(\widetilde{s}_t^v / \text{Temp})}{\sum_{i=1}^{V} \exp(\widetilde{s}_t^i / \text{Temp})}. \tag{8}$$

Here, $\widetilde{s}_t^v$ signifies the model output (logits) for the v-th token at time t, with v ranging over the set V that encompasses all tokens. The selection of tokens is governed by the temperature parameter Temp. At higher temperatures, all tokens demonstrate nearly identical probabilities; as the temperature decreases, the predicted $\widetilde{s}_t^v$ progressively influences the probability of the v-th token. With a lower temperature, the probability of the token with the highest $\widetilde{s}_t^v$ approaches 1.

A.3 Details of Discriminator

A bidirectional LSTM (Bi-LSTM) serves as a powerful discriminator to assess the authenticity and quality of SMILES strings at the token-level. Unlike traditional LSTMs, the Bi-LSTM processes input SMILES strings in both forward and backward directions, enabling a comprehensive understanding of contextual information.

The calculations in a Bi-LSTM involve two separate LSTM networks, one processing the sequence from the beginning to the end (forward LSTM), and the other processing it in reverse (backward LSTM). The hidden states from both LSTMs are concatenated at each time step, providing a more nuanced representation of the sequence. The calculation is as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ii} s_t + \mathbf{W}_{hi} \overrightarrow{\boldsymbol{h}}_{t-1} + \mathbf{b}_{ii} + \mathbf{b}_{hi}), \tag{9}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{if} s_t + \mathbf{W}_{hf} \overrightarrow{\boldsymbol{h}}_{t-1} + \mathbf{b}_{if} + \mathbf{b}_{hf}), \tag{10}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{io} s_t + \mathbf{W}_{ho} \overrightarrow{\boldsymbol{h}}_{t-1} + \mathbf{b}_{io} + \mathbf{b}_{ho}), \tag{11}$$

$$\widetilde{\mathbf{c}}_t = \tanh(\mathbf{W}_{ic} s_t + \mathbf{W}_{hc} \overrightarrow{\boldsymbol{h}}_{t-1} + \mathbf{b}_{ic} + \mathbf{b}_{hc}), \tag{12}$$

$$\mathbf{c}_t = \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \widetilde{\mathbf{c}}_t, \tag{13}$$

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{c}_t). \tag{14}$$

The backward LSTM computes hidden states ($\overleftarrow{\boldsymbol{h}}_t$) using similar equations with parameters specific to the backward pass. Finally, the hidden states from both LSTMs are concatenated to obtain the final hidden state $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t]$, which is used for discrimination purposes. This Bi-LSTM approach facilitates a more robust understanding of the sequential information, enhancing the discriminator's ability to assess the quality and authenticity of the input SMILES strings.

B MCTS-based RL and Actor-critic RL Algorithms

B.1 MCTS-based RL Algorithm

MCTS is an iterative planning procedure that constructs a decision tree one rollout at one time. Each rollout consists of four stages:

- **Selection:** Beginning at the root node, the algorithm iteratively chooses the child node with the highest exploration-exploitation score until it reaches node that still has untried actions.
- **Expansion:** One of these untried actions is expanded as a new child node.
- **Simulation:** From the newly constructed node, a policy-guided rollout is executed to a terminal node to obtain an estimated reward.
- **Backpropagation:** The obtained reward is backpropagated along the visited path, updating the estimated value for every node encountered.

B.2 Actor-critic RL Algorithm

We integrate the GAN into an actor-critic RL framework: the generator functions as the actor, proposing the next SMILES token, while the discriminator serves as the critic, evaluating that token and returning feedback to update the actor. Formally, the partially generated SMILES sub-string constitutes the state, and the next token corresponds to the action. The complete objective function is presented in the model section.

The iterative tree-building process makes MCTS increasingly computationally costly as the action space widens. In contrast, InstGAN employs an actor-critic RL framework that learns a policy end-to-end from both token-level and molecular-level rewards, eliminating explicit rollouts and scaling well to larger molecular datasets.

C Chemical Properties, Evaluation Metrics and Hyperparameters

C.1 Details of Chemical Properties

Drug-likeness (QED) calculation. The QED score is determined through a weighted average of eight molecular descriptors: 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 QED can be calculated as follows:

$$\mathrm{QED} = \exp\left(\frac{\sum_{i=1}^{8} W_i \ln d_i}{\sum_{i=1}^{8} W_i}\right), \tag{15}$$

where d_i denotes the desirability function, and W_i represents the i-th descriptor.

Solubility (logP) calculation. Solubility is computed using logP, where P is the partition coefficient defined as the ratio of the concentrations of a molecule on an octanol-water surface. The logP can be calculated as follows:

$$\text{logP} = \log \frac{c_o}{c_w}, \tag{16}$$

where c_o and c_w represent the organic and water-phase substance activity, respectively.

Synthesizability (SA) calculation. Typically, Synthesizability is computed using the SA score, which assesses the ease of synthesizing a molecule. The calculation is shown as below:

$$\text{SA} = r_s - \sum_{i=1}^{5} p_i, \tag{17}$$

where r_s represents the synthetic molecule's knowledge analysis, which is the ratio of the summed fragments' contributions to all fragment numbers within the molecule, utilizing experimental results from the literature [Li and Yamanishi, 2023]. $p_i (i \in 1, ..., 5)$ is the ring complexity, stereo complexity, macrocycle penalty, size penalty, and bridge penalty.

Dopamine Receptor D2 (DRD2) calculation. The DRD2 score quantifies the likelihood of interaction between a molecule and a target protein [Olivecrona *et al.*, 2017]. We selected DRD2 as the target protein and utilized a random forest classifier from the Scikit-learn toolkit for computation. Specifically, we initially gathered DRD2-molecule interaction data from the ChEMBL30 dataset (accessed on 2022-03-17) [Gaulton *et al.*, 2012]. Subsequently, we employed the Ki values (unit: nmol/L) from ChEMBL217 as the interaction data. After eliminating missing values and duplicate molecules, we obtained a dataset of 6652 molecule-Ki pairs. To construct the model, we trained 500 decision-tree classifiers using the random forest classifier for binary classification. The input to the classifier was the SMILES representation of a molecule, which was then transformed into a 2048-dimensional ECFP4 (extended connectivity fingerprint, up to four bonds). The labels for the binary classifier were defined as follows:

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

Then, the classifier's output is determined by computing the predicted probability associated with the label of 1, representing the DRD2 score.

Note that higher scores for these properties indicate greater desirability. The QED score ranges from 0 to 1, and the logP and SA scores range from 0.1 to 1. The RDKit tool was utilized for implementation.

C.2 Details of Evaluation Metrics

Following the evaluation criteria in [Li and Yamanishi, 2023], we consider the validity, uniqueness, novelty, total, and diversity.

Validity. Validity is the ratio of chemically valid SMILES to all generated SMILES. The calculation is as follows:

$$\text{Validity} = \frac{N_{\mathcal{D}_V}}{N_{\mathcal{D}_G}}, \tag{18}$$

where $N_{\mathcal{D}_G}$ is the number of generated SMILES-like strings, and $N_{\mathcal{D}_V}$ is the number of generated chemically valid SMILES strings.

Uniqueness. Uniqueness is determined by the ratio of chemically valid and unique SMILES strings to the number of valid SMILES strings, which is calculated by

$$\text{Uniqueness} = \frac{N_{\mathcal{D}_U}}{N_{\mathcal{D}_V}}, \tag{19}$$

where $N_{\mathcal{D}_U}$ denotes the number of generated chemically valid and unique SMILES strings.

Novelty. Novelty is the ratio of chemically valid, unique, and previously unseen SMILES strings to all chemically valid and unique SMILES. The mathematical definition is as follows:

$$\text{Novelty} = \frac{N_{\mathcal{D}_N}}{N_{\mathcal{D}_U}}, \tag{20}$$

where $N_{\mathcal{D}_N}$ is the number of generated chemically valid and unique SMILES strings that are absent from the training dataset.

Total. The Total score is defined as the ratio of novel molecules to all generated ones. The formula is as follows:

$$\text{Total} = \frac{N_{\mathcal{D}_N}}{N_{\mathcal{D}_G}}. \tag{21}$$

Diversity. The diversity is also used to evaluate the *diversity* of novel SMILES strings $\mathcal{D}_N$, which is computed based on Tanimoto coefficient [Rogers and Tanimoto, 1960] of Morgan fingerprint [Cereto-Massagué *et al.*, 2015]. The mean diversity $Div(\mathcal{M}_N)$ and Tanimoto similarity $Sim(m_i, m_j)$ are defined as follows:

$$\text{Div}(\mathcal{M}_N) = 1 - \frac{1}{|\mathcal{M}_N|} \sum_{m_i, m_j \in \mathcal{M}_N} \text{Sim}(m_i, m_j), \tag{22}$$

and

$$\text{Sim}(m_i, m_j) = \frac{|m_i \& m_j|}{|m_i| + |m_j| - |m_i \& m_j|}. \tag{23}$$

Here, the scores are calculated based on the Morgan fingerprints (m_i and m_j) of two arbitrarily generated novel molecules. The resulting scores range from 0 to 1, with higher values indicating better diversity. The RDKit tool was utilized for the implementation.

C.3 Hyperparameters

For pre-training, the learning rate was set to 2e-4. InstGAN was pre-trained with up to 60,000 steps, the batch size was 1024, the MIE trade-off β was set to 1e-3, and the GR weight W^D was 7.1e-5. For multi-property optimization, QED, logP, and SA were used as the desired chemical properties, and the weights of W_{C_i} were set to 0.4, 0.3, and 0.3. The learning rate was set to 5e-6, and the GAN was trained with a maximum of 5,000 steps with a batch size of 256. Additionally, the dropout rate was 0.1, and the gradient clippings range was from -0.1 to 0.1. The MIE trade-off β was set to 2.2e-2, the GR weight W^D was 1e-5, and the smoothing factor α was set to 0.9. The L2 coefficient for the critics was 1e-6. Evaluations were conducted every 100 steps, with 10,000 samples generated. All experiments were performed within the NVIDIA GeForce RTX 3090 GPU environment.

D Average of Multiple Pretraining

	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	Diversity ↑	QED ↑	logP ↑	SA ↑
1	96.25	98.64	99.68	0.90	0.76	0.62	0.84
2	94.67	98.69	99.73	0.90	0.79	0.63	0.84
3	95.42	98.55	99.73	0.90	0.75	0.62	0.83
Average	95.45	98.63	99.71	0.90	0.77	0.62	0.84

Table D.1: Average results obtained from multiple pretraining of InstGAN.

E Details of Property Optimization

	Chemical Property	Top-1	Top-5	Top-10	Top-100	Top-1000
InstGAN	Multi-property (QED)	0.95	0.95	0.95	0.94	0.93
	Multi-property (logP)	1.0	1.0	1.0	0.97	0.89
	Multi-property (SA)	1.0	1.0	1.0	1.0	0.99
	Single-property (QED)	0.95	0.95	0.95	0.95	0.94
	Single-property (logP)	1.0	1.0	1.0	1.0	1.0
	Single-property (SA)	1.0	1.0	1.0	1.0	1.0

Table E.1: Multi- and Single-property assessment of the top-k generated molecules using InstGAN.

Figure E.1 shows the Top-1 generated molecular structures by InstGAN in single-property optimization exhibited the highest QED, logP, and SA scores.

F Discussion of Conflicting Properties

The sharpest trade-off we observe is Between QED and logP: Table G.2 shows that optimizing logP drives QED down substantially. Tables G.1-G.4 reveal only a weak negative correlation between logP and SA, while SA and QED correlate positively.

These contrasting correlations (a strong trade-off between QED and logP, weak conflict between logP and SA, and synergy between SA and QED) highlight a key insight for drug design: accessible chemical space is constrained not just by individual properties but by their interplay. Explicitly modeling and navigating these trade-offs and synergy would be a promising research direction.

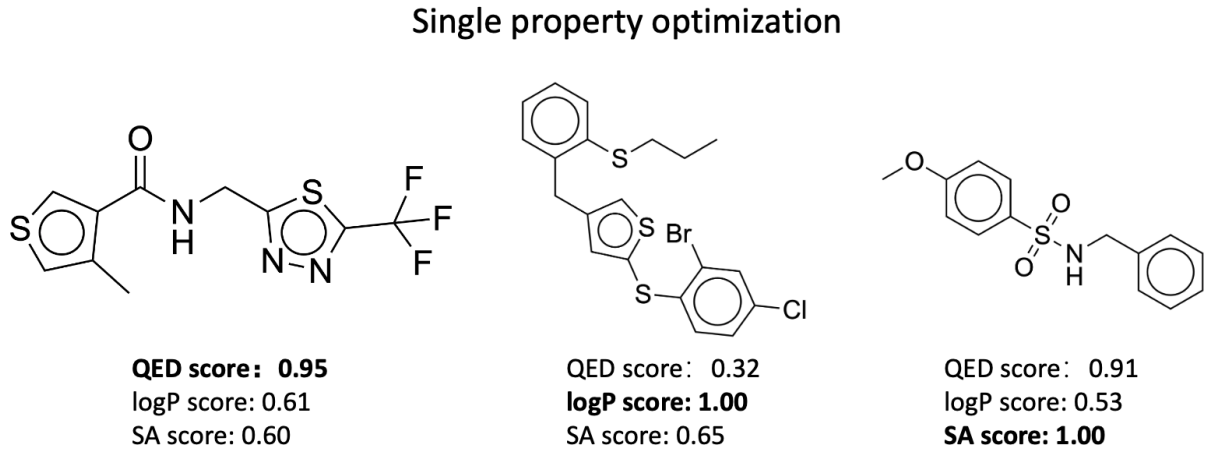


Figure E.1: The Top-1 generated molecular structures by InstGAN in single-property optimization exhibited the highest QED, logP, and SA scores.

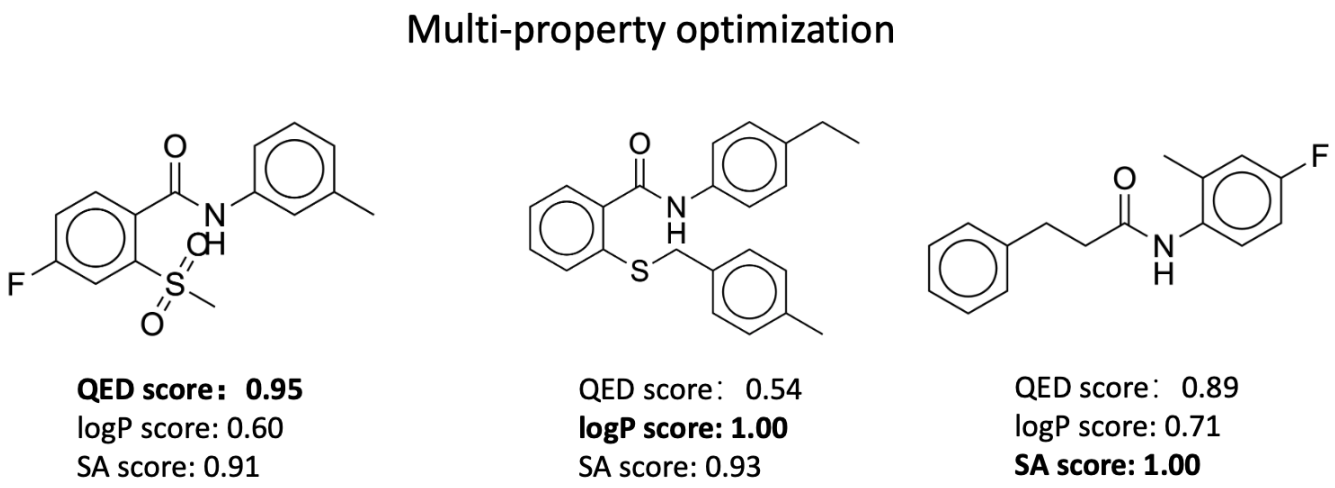


Figure E.2: The Top-1 molecular structures generated by InstGAN in multi-property optimization exhibited the highest QED, logP, and SA scores.

G Details of Ablation Studies

Tables G.1, G.2, G.3, and G.4 display the effects of λ on the performance.

H Details of Case Studies

Table H.1 assesses the performance of QED and DRD2 properties, demonstrating that the QED and DRD2 scores change with the corresponding weights. Furthermore, InstGAN enhanced the bioactivity of the generated molecules to 97.21%. Additionally, we selected a QED and DRD2 weight ratio of (0.3, 0.7) and generated bioactive molecules. Figure 4 compares the generated molecules with high DRD2 scores to similar approved drugs. These Top-3 molecules generated by InstGAN have high QED and DRD2 scores and are highly similar to approved drugs in the ChEMBL database, proving the effectiveness of InstGAN.

References

[Cereto-Massagué *et al.*, 2015] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

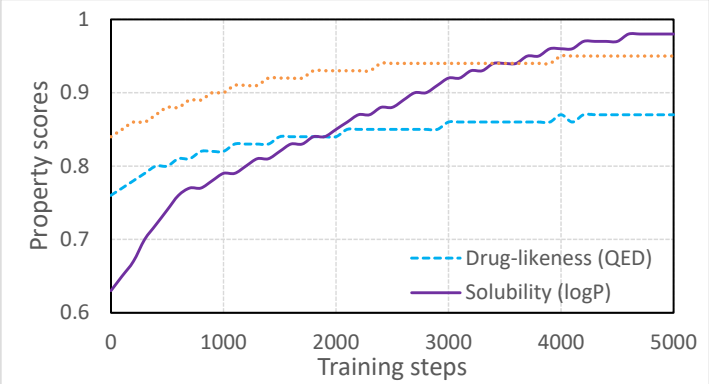


Figure E.3: Trends in chemical properties of molecules generated during single-property optimization training.

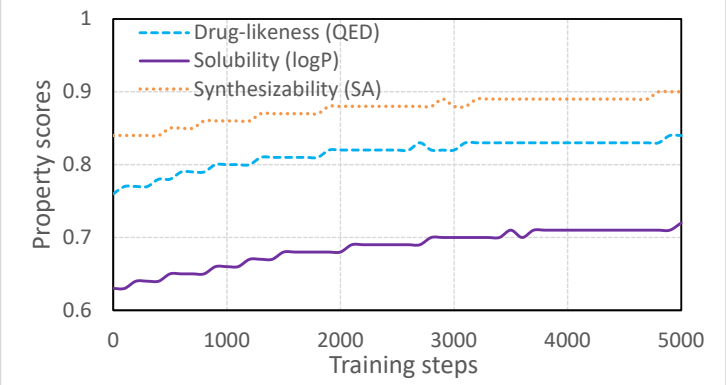


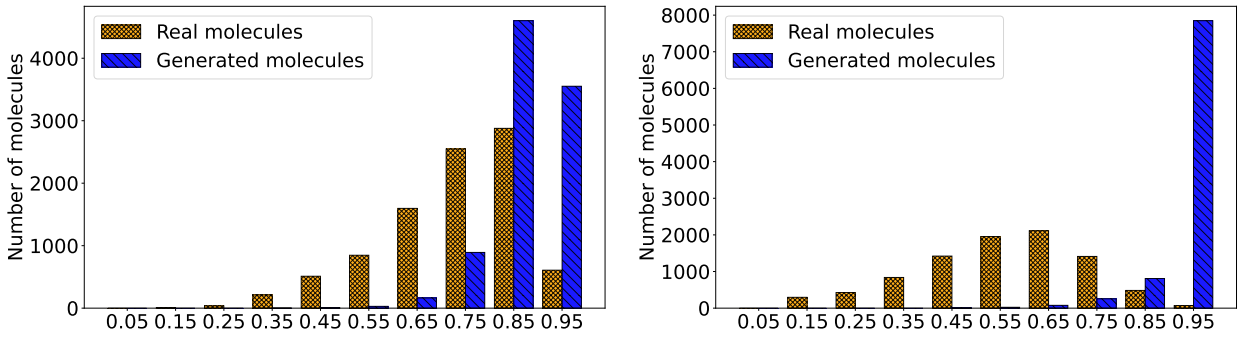
Figure E.4: Trends in chemical properties of molecules generated during multi-property optimization training.

λ	QED ↑	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	Diversity ↑	logP ↑	SA ↑	Time (h) ↓
0.0	0.77	94.73	98.89	99.72	0.90	0.60	0.81	3.73
0.1	0.79	95.62	99.11	99.83	0.90	0.60	0.81	3.89
0.2	0.81	97.05	98.76	99.74	0.90	0.60	0.82	3.87
0.3	0.83	97.89	98.31	99.69	0.89	0.60	0.83	3.83
0.4	0.85	97.96	97.24	99.55	0.89	0.61	0.84	3.85
0.5	0.87	98.52	94.24	99.56	0.88	0.61	0.86	3.81
0.6	0.87	98.90	91.87	99.57	0.88	0.61	0.87	3.71
0.7	0.87	98.69	92.09	99.57	0.88	0.62	0.87	3.56
0.8	0.86	98.61	92.01	99.67	0.88	0.62	0.88	3.38
0.9	0.86	98.53	92.01	99.56	0.88	0.62	0.88	2.91
1.0	0.86	98.34	91.95	99.61	0.88	0.62	0.88	2.79

Table G.1: Effect of λ on the performance of single-property optimization (QED) for the ZINC dataset.

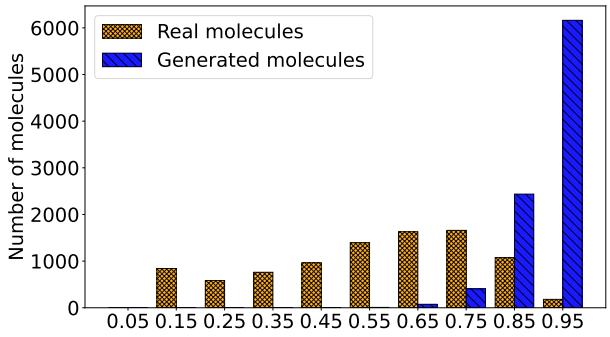
λ	logP ↑	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	Diversity ↑	QED ↑	SA ↑	Time (h) ↓
0.0	0.63	96.26	98.70	99.73	0.90	0.76	0.84	3.49
0.1	0.66	95.06	99.16	99.78	0.90	0.76	0.81	3.53
0.2	0.71	95.45	99.43	99.83	0.90	0.74	0.82	3.39
0.3	0.78	95.78	99.32	99.87	0.90	0.71	0.82	3.50
0.4	0.90	96.36	97.77	99.97	0.89	0.62	0.82	3.43
0.5	0.97	97.18	92.80	100.00	0.88	0.50	0.78	3.45
0.6	0.95	94.71	95.20	99.94	0.89	0.52	0.78	3.03
0.7	0.93	94.42	95.92	99.98	0.90	0.53	0.78	2.74
0.8	0.91	92.35	97.86	99.97	0.90	0.57	0.79	2.50
0.9	0.88	91.59	98.50	99.97	0.90	0.61	0.81	2.48
1.0	0.92	93.18	97.36	99.93	0.90	0.55	0.80	2.41

Table G.2: Effect of λ on the performance of single-property optimization (logP) for the ZINC dataset.



(a) Drug-likeness (QED) distributions.

(b) Solubility (logP) distributions.



(c) Synthesizability (SA) distributions.

Figure E.5: Property distributions of generated molecules with single-property optimization.

λ	SA ↑	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	Diversity ↑	QED ↑	logP ↑	Time (h) ↓
0.0	0.84	96.20	98.73	99.72	0.90	0.76	0.63	3.80
0.1	0.84	96.20	98.74	99.72	0.90	0.76	0.63	3.79
0.2	0.85	97.05	98.50	99.61	0.89	0.78	0.62	3.77
0.3	0.88	97.96	97.07	99.62	0.89	0.79	0.62	3.75
0.4	0.92	98.77	91.89	99.48	0.88	0.80	0.63	3.82
0.5	0.92	98.55	92.69	99.47	0.87	0.79	0.62	3.44
0.6	0.91	98.72	93.25	99.52	0.88	0.79	0.62	2.73
0.7	0.92	98.52	92.15	99.47	0.87	0.79	0.63	2.29
0.8	0.91	98.27	94.03	99.49	0.88	0.78	0.62	2.06
0.9	0.91	98.39	93.43	99.55	0.88	0.79	0.63	1.94
1	0.91	98.35	93.25	99.54	0.88	0.79	0.63	1.91

Table G.3: Effect of λ on the performance of single-property optimization (SA) for the ZINC dataset.

[Gaulton *et al.*, 2012] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2012.

[Guo and Schwaller, 2023] Jeff Guo and Philippe Schwaller. Augmented memory: Capitalizing on experience replay to accelerate de novo molecular design. *arXiv preprint arXiv:2305.16160*, 2023.

[Li and Yamanishi, 2023] Chen Li and Yoshihiro Yamanishi. SpotGAN: A reverse-transformer gan generates scaffold-constrained molecules with property optimization. In *Proceedings of the Joint European Conference on Machine Learning*

λ	QED ↑	logP ↑	SA ↑	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	Diversity ↑	Time (h) ↓
0.0	0.76	0.63	0.84	95.89	98.83	99.68	0.90	4.36
0.1	0.76	0.63	0.84	95.89	98.83	99.68	0.90	4.43
0.2	0.79	0.64	0.83	96.93	98.90	99.81	0.90	4.46
0.3	0.80	0.65	0.85	97.46	98.63	99.71	0.89	4.48
0.4	0.82	0.68	0.87	98.25	97.14	99.55	0.88	4.50
0.5	0.84	0.72	0.90	98.96	92.25	99.50	0.87	4.19
0.6	0.84	0.72	0.91	99.18	91.54	99.66	0.87	4.14
0.7	0.83	0.72	0.91	98.72	91.73	99.54	0.87	3.67
0.8	0.83	0.72	0.92	98.70	91.60	99.56	0.87	3.38
0.9	0.82	0.70	0.91	98.49	92.65	99.54	0.88	2.92
1.0	0.82	0.70	0.91	98.40	93.42	99.53	0.88	2.78

Table G.4: Effect of λ on the performance of multi-property optimization for the ZINC dataset.

Weight (QED, DRD2)	QED ↑	DRD2 ↑	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑
Pretrain	0.70	0.21	94.87	98.46	99.13
(0.5, 0.5)	0.80	0.27	97.21	87.52	99.14
(0.4, 0.6)	0.78	0.28	97.17	84.29	99.05
(0.3, 0.7)	0.76	0.29	96.66	84.80	99.24
(0.2, 0.8)	0.73	0.30	96.25	84.35	99.47
(0.1, 0.9)	0.70	0.31	95.52	85.49	99.44

Table H.1: Effect of weights on QED and DRD2 property scores.

and Knowledge Discovery in Databases, pages 323–338. Springer, 2023.

[Li *et al.*, 2022] Chen Li, Chikashige Yamanaka, Kazuma Kaitoh, and Yoshihiro Yamanishi. Transformer-based objective-reinforced generative adversarial network to generate desired molecules. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3884–3890, 2022.

[Olivecrona *et al.*, 2017] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):1–14, 2017.

[Rogers and Tanimoto, 1960] David J Rogers and Taffee T Tanimoto. A computer program for classifying plants. *Science*, 132(3434):1115–1118, 1960.