InstGAN: Instant Actor-Critic-Driven GAN for De Novo Molecule Generation and Property Optimization

Huidong Tang[1,2] , **Chen Li**[3] , **Sayaka Kamei**[2] , **Yoshihiro Yamanishi**[4] and **Yasuhiko Morimoto**[2]
[1]Cloud Computing Technology and Application Department, Shandong Institute of Commerce and Technology, Jinan, China
[2]Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima, Japan
[3]D3 Center, The University of Osaka, Osaka, Japan
[4]Graduate School of Informatics, Nagoya University, Nagoya, Japan
tanghd24@163.com, li.chen.d3c@osaka-u.ac.jp, s10kamei@hiroshima-u.ac.jp,
yamanishi@i.nagoya-u.ac.jp, morimo@hiroshima-u.ac.jp

Abstract

Deep generative models, such as generative adversarial networks (GANs), have been employed for *de novo* molecular generation in drug discovery. Most prior studies have utilized reinforcement learning (RL) algorithms, particularly Monte Carlo tree search (MCTS), to handle the discrete nature of molecular representations in GANs. However, due to the inherent instability in training GANs and RL models, along with the high computational cost associated with MCTS sampling, MCTS RL-based GANs struggle to scale to large chemical databases. To tackle these challenges, this study introduces a novel GAN based on actor-critic RL with instant and global rewards, called InstGAN, to generate molecules at the token-level with multi-property optimization. Furthermore, maximized information entropy is leveraged to alleviate the mode collapse. The experimental results demonstrate that InstGAN outperforms other baselines, achieves comparable performance to state-of-the-art models, and efficiently generates molecules with multi-property optimization. The code is available at: https://github.com/tang777777/InstGAN.

1 Introduction

Modern human healthcare and well-being are intricately intertwined with the field of drug discovery, which seeks to uncover new chemical compounds with therapeutic effects. However, traditional drug discovery is a time-consuming and expensive endeavor, taking an average of 12 years and costing 2.6 billion USD [Chan *et al.*, 2019]. To expedite the process and mitigate costs, artificial intelligence (AI) has garnered the attention of the pharmaceutical industry [Paul *et al.*, 2021]. Among the recent applications of AI, deep generative models have demonstrated remarkable progress, as exemplified by DALL·E2 in the realm of computer vision and ChatGPT in natural language processing (NLP) [OpenAI, 2023]. The adoption of such models has also become increasingly prominent in the field of drug discovery [Chen *et al.*, 2018].

Molecular graphs [Shi *et al.*, 2020] and simplified molecular input line entry systems (SMILES) strings [Weininger, 1988] constitute the two primary representations of molecules in deep generative models. However, generating molecules with desired chemical properties using such discrete representations is a non-trivial task. Most prior studies related to generative adversarial networks (GANs) [Yu *et al.*, 2017; Guimaraes *et al.*, 2017; De Cao and Kipf, 2018] typically update the generator by integrating the output probability of the discriminator with the chemical properties of generated molecules as a reward for reinforcement learning (RL), following the REINFORCE algorithm [Williams, 1992]. Due to the inability of GANs to calculate rewards for partially generated molecules, Monte Carlo tree search (MCTS) is frequently utilized for sampling and completing molecules [Li *et al.*, 2022]. Unfortunately, the integration of RL algorithms with GANs further exacerbates the instability of the training process. Achieving training stability with MCTS demands a substantial number of samples, rendering the process highly time-consuming [Li and Yamanishi, 2023].

Furthermore, most aforementioned studies on *de novo* molecular generation have focused on optimizing a single chemical property. However, in practical applications, it is often desirable to generate molecules that satisfy multiple chemical property constraints. In contrast to the former, multi-property optimization is highly complicated and challenging to achieve in nature. This is because the multi-property optimization task entails not only learning the semantic and syntactic rules of molecules to generate valid molecules from scratch but also finding pathways to optimize the distribution of chemical properties during the process [Barshatski *et al.*, 2021]. For example, molecules exhibiting both drug-likeness and dopamine receptor (DRD2) activity represent only 1.6% of the generated molecules [Jin *et al.*, 2019]. Therefore, employing deep generative models for *de novo* molecular generation with multi-property optimization holds significance to the drug discovery industry [Barshatski and Radinsky, 2021].

Inspired by the previous studies in [De Cao and Kipf, 2018; Tang *et al.*, 2023], this study introduces a novel GAN based on actor-critic RL with instant rewards (IR) and global rewards (GR), called InstGAN, to generate molecules at the token-level with multi-property optimization. Specifically, the generator is constructed using a long short-term memory (LSTM) that generates SMILES strings in an autoregressive manner. The discriminator quantifies each token based on SMILES substrings produced by the generator. To enhance the ability of the discriminator to quantize tokens, a bidirectional LSTM (Bi-LSTM) is chosen as the discriminator. Additionally, multi-property prediction networks with the same structure as the discriminator predict the corresponding property scores for each token as well. Subsequently, the scores of discriminator and property prediction networks, along with their global-level scores, serve as rewards for RL. To expedite the training process and facilitate its application to extensive chemical databases, an IR calculation based on actor-critic RL is proposed to update the generator. Furthermore, the maximized information entropy (MIE) is included in the generator loss function to mitigate mode collapse and enhance molecular diversity. The main contributions are

- **Novel reward calculation:** This study proposes an actor-critic RL-based approach to calculate IR and GR for molecular generation with multi-property optimization.
- **Scalability for chemical property optimization:** InstGAN exhibits versatile scalability, enabling flexible expansion from single-property to arbitrary multi-property optimized molecular GAN.
- **Superior performance:** Experimental results validate that InstGAN outperforms other baselines, achieves comparable performance to state-of-the-art (SOTA) models, and demonstrates the ability to generate molecules with multi-property optimization in a fast and efficient manner.

2 Related Work

2.1 Variational Autoencoder (VAE)-based Models

Two VAE variants, Character-VAE and Grammar-VAE [Kusner *et al.*, 2017], integrate parse trees with VAEs to facilitate the generation of syntactically valid molecules. However, due to ignoring the semantic information of the molecular representations, the correlation between the training set and the generated molecules cannot be guaranteed. In contrast, Syntax-VAE [Dai *et al.*, 2018] ensures that the generated molecules are both syntactically valid and semantically meaningful. JT-VAE [Jin *et al.*, 2018] adopts a two-step VAE approach, first generating tree-structured scaffolds based on chemical substructures and then combining these outputs into complete molecules using a graph message-passing network [Dai *et al.*, 2016; Gilmer *et al.*, 2017]. Nonetheless, a major limitation of VAEs is the typically limited size of the latent space, which may restrict the capacity to produce highly realistic molecules.

2.2 Flow-based Models

GraphAF [Shi *et al.*, 2020] utilizes a flow-based autoregressive model to generate molecular graphs. GraphDF [Luo *et al.*, 2021] incorporates invertible modulo shift transforms to connect discrete latent variables with graph nodes and edges, resulting in the generation of molecular graphs. MoFlow [Zang and Wang, 2020] adopts a Glow-based model [Kingma and Dhariwal, 2018] to generate chemical bonds as graph edges and employs a graph conditional flow to subsequently generate atoms as graph nodes, followed by posthoc validity correction. GraphCNF [Lippe and Gavves, 2021] falls under the category of flow-based models, commonly applied in various data domains. In molecular graph generation, GraphCNF leverages flow-based techniques to address the unique challenges associated with generating molecular graphs. However, flow-based models exhibit several notable limitations: the computation of the Jacobian matrix is time-consuming, often requiring approximations, and the necessity for network invertibility imposes constraints on their representational capacity, limiting their flexibility in modeling complex, high-dimensional data distributions [Zhang *et al.*, 2021]. Additionally, ensuring invertibility across all layers can lead to challenges in model training and optimization.

2.3 Diffusion-based Models

Recently, diffusion models have found application in the domain of molecular generation, where they are being utilized to tackle the intricate challenges associated with generating molecular structures that adhere to specific chemical and property constraints. DiGress [Vignac *et al.*, 2023] iteratively refines noisy graphs by adding or removing edges and adjusting categories, which results in the generation of molecular graphs with node and edge attributes suitable for classification. GDSS [Jo *et al.*, 2022] and D2L-OMP [Guo *et al.*, 2023] are the two SOTA models in the realm of molecular generation. GDSS skillfully captures the joint distribution of molecular nodes and edges, generating molecular replicas closely aligned with the training distribution. D2L-OMP generates molecules with property optimization based on a hybrid Gaussian distribution by employing diffusion on two structural levels: molecules and molecular fragments. This hybrid Gaussian distribution is then utilized in the reverse denoising process.

2.4 GAN-based Models

SeqGAN [Yu *et al.*, 2017] pioneered the incorporation of MCTS-based RL into GAN architecture, specifically designed to handle discrete text. This innovation has inspired various studies on molecular generation using GANs. MolGAN [De Cao and Kipf, 2018] introduces an implicit, likelihood-free discrete GAN for generating small molecular graphs. However, MolGAN faces an overfitting problem, leading to less than 5% uniqueness in the generated molecules. ORGAN [Guimaraes *et al.*, 2017] integrates domain-specific knowledge as rewards for generating SMILES strings through the MCTS-based RL algorithm. TransORGAN [Li *et al.*, 2022] leverages a transformer architecture to capture semantic information and employs variant SMILES technique for syntax rule learning. SpotGAN [Li and Yamanishi, 2023] adopts a first-decoder-then-encoder transformer model for generating SMILES strings from a given scaffold. However, the use of MCTS-based RL in GANs often demands a substantial number of samples for training stability, making it impractical for extensive chemical databases. EarlGAN [Tang *et al.*, 2023], while capable of generating valid molecules on large chemical databases, lacks the ability to optimize chemical properties, particularly multiple properties simultaneously.

This study introduces an actor-critic RL-driven GAN that employs IR and GR to enhance the efficiency of learning semantic and syntactic rules in SMILES strings. Furthermore, our other goal is to optimize molecule generation across diverse chemical properties.

3 Model

Figure 1 illustrates the overall architecture of InstGAN and highlights the three key substructures (i.e., the generator, discriminator, and multiple chemical property prediction networks) that are crucial for generating molecules with multi-property optimization from scratch. Formally, let $G_{\boldsymbol{\theta}}$ and $D_{\boldsymbol{\phi}}$ represent the generator and discriminator of InstGAN with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. $\boldsymbol{S}_{1:T} = [s_1, \cdots, s_T]$ denotes a SMILES string with a sequence length of T. Then, the min-max optimization procedure [Goodfellow *et al.*, 2014] is implemented as follows:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} V(G_{\boldsymbol{\theta}}, D_{\boldsymbol{\phi}}) = \\ \mathbb{E}_{\boldsymbol{S} \sim p_r(\boldsymbol{S})}[\log D_{\boldsymbol{\phi}}(\boldsymbol{S})] + \mathbb{E}_{\boldsymbol{S} \sim p_{\boldsymbol{\theta}}(\boldsymbol{S})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{S}))], \quad (1)$$

where $p_r(\cdot)$ and $p_{\boldsymbol{\theta}}(\cdot)$ represent the distributions of training SMILES strings and generated sequences, respectively. Following a similar approach to NLP methods [de Masson d'Autume *et al.*, 2019; Fedus *et al.*, 2018], InstGAN utilizes an autoregressive generator and discriminator. Notably, this design allows for the allocation of dense rewards at the token-level. For detailed explanation of InstGAN's design motivations and a full description of its generator and discriminator architectures, see Appendix A[1].

3.1 Token-level Critics

Unlike graph-based approaches, molecular generative models relying on SMILES strings often face challenges in ensuring high validity due to the intricacies of checking valence during the autoregressive generation process. Typically, invalid SMILES strings may arise from mismatched tokens, requiring the removal or replacement of other suitable tokens to restore validity. A crucial aspect of addressing this issue lies in the meticulous assessment of SMILES strings at the token level. We therefore integrate the validity assessment directly into the generation process, by treating each token generation as an action in a token-level actor-critic RL loop: at every step, the generator (actor) proposes a candidate token, while the discriminator (critic) assess its syntactic validity and return a dense reward. This feedback system ensures the generator of high-validity token generation.

Token-level discriminator. In InstGAN, the generator employs an autoregressive process to produce SMILES strings, serving as inputs for the discriminator. Diverging from traditional discriminators, InstGAN performs token-level discrimination for each generated token. Specifically, with a Bi-LSTM, let $\overrightarrow{\boldsymbol{S}}_{1:t}$ and $\overleftarrow{\boldsymbol{S}}_{t:T}$ represent the forward and backward SMILES substrings, respectively. The discriminator calculates the probability r_t^D that it deems the t-th token $\widetilde{s}_t$ of the SMILES string as true. The calculation is as follows:

$$r_t^D = D_{\boldsymbol{\phi}}(\widetilde{s}_t | \overrightarrow{\boldsymbol{S}}_{1:t}, \overleftarrow{\boldsymbol{S}}_{t:T}). \quad (2)$$

To assess the validity of the entire SMILES string, we calculate the global discriminator score as

$$r_{1:T}^D = \frac{1}{T} \sum_{t=1}^{T} r_t^D. \quad (3)$$

Token-level property prediction networks. The chemical property prediction networks are replicated from the discriminator and share the same structure. In the pre-training phase, the entire SMILES string serves as input, and the networks utilize the corresponding property values of the complete SMILES string as labels for the chemical properties of its tokens at each step. In the training phase of InstGAN, these networks calculate both instant and global chemical properties of the generated SMILES string. Collaborating with the discriminator, they contribute to updating the generator using actor-critic RL-based rewards.

3.2 Instant and Global Reward Calculation

Following the actor-critic RL algorithm, the generator functions as the actor network responsible for action selection, while the discriminator, along with the property prediction networks, serves as the critic for reward calculation. However, in contrast to the traditional actor-critic RL algorithm [Konda and Tsitsiklis, 1999], which calculates the reward at the last time step T, we compute the reward for each token as

$$R_t^D = 2r_t^D - 1 + W^D r_{1:T}^D, \quad (4)$$

where W^D represents the weight assigned to the discriminator's GR. Similarly, the reward of the property prediction networks can be calculated and denoted as $R_t^{C_i}$, $i \in [1, 2, \cdots, N]$, and N indicates the number of chemical properties to be optimized. Subsequently, the overall reward for chemical properties is represented as $R_t^C = \sum_{i=1}^{N} W_{C_i} R_t^{C_i}$, where W_{C_i} denotes the weight assigned to the i-th chemical property prediction network. Finally, the total reward R_t can be calculated as follows:

$$R_t = (1 - \lambda) R_t^D + \lambda R_t^C, \quad (5)$$

where λ represents a hyperparameter that balances the trade-off between the GAN and RL.

3.3 Objective Function

In the iterative adversarial process, the generator is updated by the MCTS-based RL algorithm through the sampling of numerous samples, leading to a computational training process. In contrast, InstGAN utilizes rewards derived from the actor-critic RL algorithm and MIE for the calculation. Please see the description of the MCTS-based RL and actor-critic RL for algorithms comparison in Appendix B. The overall loss function for the generator is calculated as

$$\mathcal{L}_{\boldsymbol{\theta}} = \mathcal{L}_{RL} + \beta \mathcal{L}_{MIE}, \quad (6)$$

[1] The Appendix is available at: https://github.com/tang777777/InstGAN/blob/main/Appendix.pdf

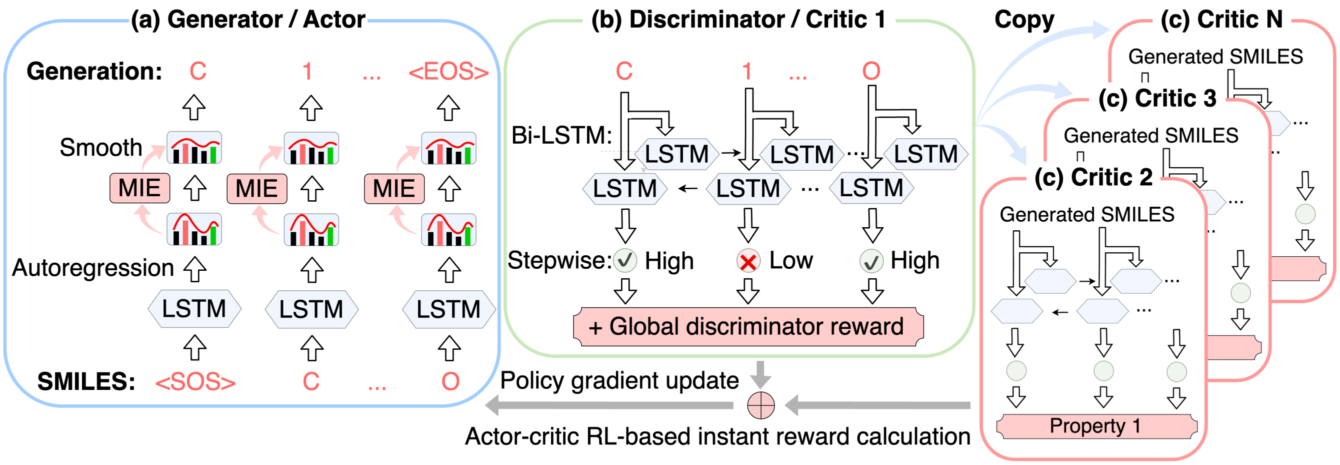


Figure 1: The architectural overview of InstGAN comprises three main substructures. **(a) The generator**, featuring an LSTM, produces tokens in an autoregressive manner at each time step. MIE is employed to enhance the likelihood of sampling different tokens, smoothing the output probability distribution for the generator. **(b) The discriminator**, a Bi-LSTM, scores generated tokens based on both forward and backward at each time step, enabling semantic and syntactic discrimination at the token-level. Higher probabilities are assigned to likely generated tokens, while those with errors receive lower probabilities. The average of all-token probabilities assesses overall generation quality from a global perspective. **(c) Multiple pretrained chemical property prediction networks**, labeled Critic 1 to Critic N, have the same structure as the discriminator. They predict various chemical properties for each token of a generated SMILES string. Similarly, the sum of stepwise scores is averaged to provide a global property score for the entire SMILES string. Finally, scores from the discriminator and critics serve as rewards in the actor-critic RL algorithm, co-updating the generator via the policy gradients.

where $\mathcal{L}_{RL}$ and $\mathcal{L}_{MIE}$ represent the loss functions of RL and MIE, respectively, with β serving as the trade-off parameter between them. In accordance with the policy gradient, $\mathcal{L}_{RL}$ is calculated by minimizing the expected reward score:

$$\mathcal{L}_{RL} = -\frac{1}{T}\sum_{\boldsymbol{S}_{1:T}} (R_t - b_t)\log p_{\boldsymbol{\theta}}(s_t|\boldsymbol{S}_{1:t-1}), \tag{7}$$

where b_t denotes the baseline using the global moving-average rewards [Sutton and Barto, 2018], which is calculated using both the mean reward $\bar{R}$ across the current batch and the previous baseline b_{t-1}, calculated as

$$b_t = (1-\alpha)\bar{R} + \alpha b_{t-1}. \tag{8}$$

Here, α denotes a smoothing factor. To encourage the generator to sample tokens with probabilities other than the highest, MIE $\mathcal{L}_{MIE}$ is incorporated into the generator's loss function. This addition serves to smooth the probability distribution, mitigating the mode collapse problem and promoting diversity in generating molecules.

$$\mathcal{L}_{MIE} = \frac{1}{T}\sum_{\boldsymbol{S}_{1:T}}\sum_{v=1}^{V} p_{\boldsymbol{\theta}}(s_t^v)\log p_{\boldsymbol{\theta}}(s_t^v), \tag{9}$$

where V represents the size of chemical vocabulary. Algorithm 1 outlines the training procedure for InstGAN. The generator, discriminator, and multiple chemical property prediction networks are first pre-trained. Then, the generator is trained to generate a dataset. Afterward, the discriminator and critics are updated using both real and generated SMILES strings. Finally, the IR and GR values are calculated to update the generator's parameters.

Dataset	MaxL	MinL	AvgL	QED	logP	SA	DRD2
ZINC	109	9	44	0.73	0.56	0.56	0.24
ChEMBL	116	10	47	0.57	0.67	0.62	0.25

* MaxL, MinL, and AvgL indicate the maximum, minimum, and average length of the SMILES strings.

Table 1: Statistical descriptions of the datasets.

4 Experiments

4.1 Experimental Setup

Datasets. The performance of InstGAN was validated through experiments on two widely used chemical datasets: ZINC [Irwin *et al.*, 2012] and ChEMBL [Gaulton *et al.*, 2017]. The ZINC dataset contains $250,000$ drug-like molecules. The ChEMBL dataset includes approximately 1.6 million molecules, with each having a median of 27 and a maximum of 88 heavy atoms.

Chemical properties. Drug-likeness (QED) quantifies the probability that a molecule belongs to a drug [Bickerton *et al.*, 2012]. **Solubility (logP)** measures how well a molecule dissolves in lipid versus aqueous environments, quantified as the Log of the partition coefficient [Comer and Tam, 2001]. **Synthesizability (SA)** is defined by the synthetic accessibility score, evaluating the ease with which a molecule can be synthesized [Ertl and Schuffenhauer, 2009]. **Dopamine receptor D2 (DRD2)** is a central nervous system G protein-coupled receptor that is essential for dopamine-mediated signaling [Olivecrona *et al.*, 2017]. Table 1 provides detailed statistical descriptions of the datasets.

Metrics. Validity is assessed by the proportion of chemically valid molecules among all generated ones, validated prac-

Algorithm 1 Training Procedure of InstGAN.

```
Data: a SMILES dataset $\mathcal{D}_{real}$
Initialization: $G_{\boldsymbol{\theta}}$, $D_{\boldsymbol{\phi}}$, $C^i_{\boldsymbol{\varphi}_i}, i \in [1, \cdots, N]$
Generate a dataset $\mathcal{D}_{fake}$ from scratch
// Pre-train the discriminator
for $k = 1 \to$ d-steps do
    Update $D_{\boldsymbol{\phi}}$ on $\mathcal{D}_{real}$ and $\mathcal{D}_{fake}$
end for
// Pre-train the generator
for $k = 1 \to$ g-steps do
    Update $G_{\boldsymbol{\theta}}$ on $\mathcal{D}_{real}$
end for
// Pre-train property networks
for $i = 1 \to N$ do
    for $k = 1 \to$ p-steps do
        Update $C^i_{\boldsymbol{\varphi}_i}$ on $\mathcal{D}_{real}$
    end for
end for
// Multi-property optimization
for $k = 1 \to$ m-steps do
    $G_{\boldsymbol{\theta}}$ updates the generated dataset $\mathcal{D}_{fake}$
    Update $D_{\boldsymbol{\phi}}$ and $C^i_{\boldsymbol{\varphi}_i}$
    $D_{\boldsymbol{\phi}}$ discriminates between $\mathcal{D}_{real}$ and $\mathcal{D}_{fake}$ and outputs the IR and GR
    $C^i_{\boldsymbol{\varphi}_i}$ calculates the IR and GR for chemical properties
    Update $G_{\boldsymbol{\theta}}$ based on the rewards
end for
```

tically using the RDKit tool [Landrum, 2013]. **Uniqueness** is determined by the proportion of non-duplicated molecules among all valid ones. **Novelty** is defined as the proportion of unique molecules not present in the training set. **Total** is the ratio of novel molecules to all generated ones. **Diversity** is calculated as the average Tanimoto distance [Rogers and Tanimoto, 1960] between the Morgan fingerprints [Cereto-Massagué *et al.*, 2015] of novel molecules. All these properties and metrics are normalized to a range of $[0, 1]$, with a higher score indicating better performance. For a detailed description of the chemical properties, evaluation metrics and hyperparameters, please refer to Appendix C.

4.2 Evaluation Results

Comparison results with baselines. Table 2 compared the results of InstGAN with various baseline models (including VAE-, flow-, diffusion-, and GAN-based models) for chemical property optimization on the ZINC dataset. To ensure a fair comparison, InstGAN was pre-trained several times, and the average results are presented. Additional details on these multiple pre-training sessions are provided in Appendix D. For VAE-based models, although RNN-Attention and TransVAE generated molecules that were highly unique and novel, their validity (i.e., $< 71.6\%$) was significantly lower compared to InstGAN. InstGAN outperformed Character-VAE and Grammar-VAE in all metrics. Although JT-VAE exhibited high validity and novelty, the uniqueness was only 19.75%, significantly lower than that of InstGAN. For flow-based models, despite exhibiting high uniqueness and novelty, their validity was lower, specifically less than 89.03%, which significantly trails behind InstGAN. InstGAN demonstrated comparable performance to SOTA diffusion models GDSS and D2L-OMP. All models, including pre-training and tasks involving single properties (QED, logP, SA), as well as multi-property tasks (QED, logP, SA), achieved an overall score surpassing 93.87%. Among GAN-based models, while MolGAN and TransORGAN exhibited a novelty of 100.0%, their validity and uniqueness fell significantly lower than InstGAN. Specifically, MolGAN demonstrated a mere 4.3% uniqueness, and TransORGAN exhibited a validity of only 74.31%. ORGAN's validity stood at 67.96%, markedly lower than InstGAN's impressively high validity exceeding 95.45%. Furthermore, InstGAN outperformed SpotGAN in terms of validity, uniqueness, novelty, and total score, with the highest overall performance. InstGAN was trained for single- and multi-property optimization. In single-property optimization, InstGAN was trained individually using QED, logP, and SA. In multi-property optimization, these three properties were used to jointly train InstGAN. The validity, uniqueness, novelty, and total score all reached up to 93.87%. Overall, while InstGAN generated molecules from less informative SMILES strings compared to graph-based models, it surpassed VAE-, flow-, and GAN-based baselines and demonstrated comparable performance to SOTA diffusion models. This highlights InstGAN's robust capability in molecular generation, demonstrating excellence in both single-property and multi-property optimization.

Property optimization. Owing to the diverse sequence representations in SMILES strings, they inherently introduce more noise compared to molecular graphs. Furthermore, as molecular graphs typically contain more detailed information, including atoms, chemical bonds, and valences, the task of molecular generation based on SMILES strings is more challenging. As illustrated in Table 2, InstGAN performed well in molecular generation in all evaluation metrics. InstGAN obtained almost 100% novelty in multi-property optimization, with validity and uniqueness exceeding 97.7%. InstGAN's ability to excel in learning both semantic and syntactic features within SMILES strings is the key contributing factor to this achievement. This achievement surpassed the capabilities of all other models that rely on SMILES strings for chemical property optimization.

Table 3 and Table E.1 show the top-k property scores for the multi- and single-property optimization of the generated molecules, respectively. In multi-property optimization, InstGAN enhanced all targeted chemical properties. The QED scores showed a 30.1% improvement (from 0.73 to 0.95) for Top-1 and 27.4% improvement (from 0.73 to 0.93) for Top-1000, compared with the training dataset. Furthermore, the generated molecules of InstGAN with logP and SA as the desired properties exhibited a notable improvement, with logP scores increasing by 78.6% for Top-1 and 58.9%, as well as 76.8% for SA scores for Top-1000, respectively. In both property optimization, InstGAN generated molecules with higher QED scores, compared with other baselines. Especially, InstGAN improved the QED score of the Top-1000 by 9.4% and 10.6%, comparing to the SOTA D2L-OMP baseline. In single-property optimization, the generated molecules of InstGAN demonstrated substantial improvements, with the scores

	Model	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	Total (%) ↑
VAE-based	RNN-Attention [Dollar *et al.*, 2021]	71.57	99.94	100.0	71.53
	TransVAE [Dollar *et al.*, 2021]	25.39	99.96	100.0	25.38
	Character-VAE [Kusner *et al.*, 2017]	86.65	81.21	26.36	18.55
	Grammar-VAE [Kusner *et al.*, 2017]	91.91	77.24	11.90	8.45
	JT-VAE [Jin *et al.*, 2018]	100.0	19.75	99.75	19.70
Flow-based	GraphAF [Shi *et al.*, 2020]	68.00	99.10	100.0	67.39
	GraphDF [Luo *et al.*, 2021]	89.03	99.16	100.0	88.28
	MoFlow [Zang and Wang, 2020]	81.76	99.99	100.0	81.75
	GraphCNF [Lippe and Gavves, 2021]	63.56	100.0	100.0	63.56
Diffusion-based	GDSS [Jo *et al.*, 2022]	97.01	99.64	100.0	96.66
	D2L-OMP [Guo *et al.*, 2023]	97.51	99.88	100.0	97.39
GAN-based	ORGAN [Guimaraes *et al.*, 2017]	67.96	98.20	98.39	65.66
	MolGAN [De Cao and Kipf, 2018]	95.30	4.30	100.0	4.10
	TransORGAN [Li *et al.*, 2022]	74.31	91.79	100.0	68.21
	SpotGAN [Li and Yamanishi, 2023]	93.26	92.78	92.75	80.25
InstGAN	Pre-train (Average)	95.45	98.63	99.71	93.87
	Property (QED)	97.89	98.31	99.69	95.94
	Property (logP)	96.65	98.42	99.93	95.05
	Property (SA)	97.46	98.59	99.75	95.85
	Multi-property	97.71	98.71	99.64	96.10

* The values in the gray cells indicate the maximum scores in the respective columns.

Table 2: Comparison results of InstGAN with various baseline models for chemical property optimization on the ZINC dataset.

Model (Property)	Top-1	Top-5	Top-10	Top-100	Top-1000
GraphAF	0.94	0.93	0.92	0.86	0.57
GDSS	0.94	0.94	0.93	0.91	0.85
MoFlow	0.93	0.92	0.92	0.87	0.78
D2L-OMP	0.95	0.94	0.94	0.91	0.85
InstGAN	0.95	0.95	0.95	0.95	0.94
InstGAN (Multi)	0.95	0.95	0.95	0.94	0.93

Table 3: QED assessment of the top-k generated molecules.

increasing by 30.1%, 78.6%, and 78.6% for Top-1, and 28.8%, 78.6%, and 78.6% for Top-1000, compared with the training dataset. Overall, InstGAN showcased substantial enhancements in property optimization, underscoring its effectiveness in improving targeted chemical properties.

Figures E.1 and E.2 in the appendix show the top-ranked (Top-1) molecular structures generated by InstGAN in the single- and multi-property optimization tasks, respectively. The generated molecule adhered to Hückel's rules [Klein and Trinajstic, 1984], essential for obtaining the targeted chemical properties of new drugs. These findings suggest that InstGAN successfully produced new drug-like molecules with relatively high QED, logP, and SA scores.

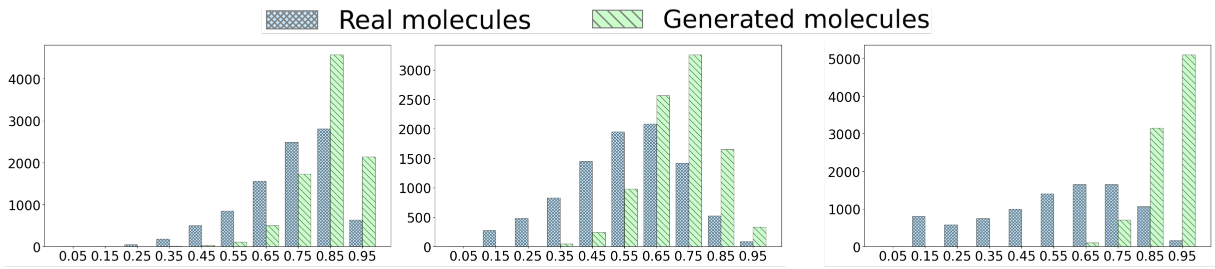
Figures E.3 and E.4 in the appendix depict the curves of average chemical property values versus training steps for molecules generated in single- and multi-property optimization tasks. Chemical properties for both single- and multi-property optimization exhibited gradual increases over 5000 training steps. In single-property optimization, the independence of the three chemical properties resulted in distinct and noticeable score increases. Moreover, during the multi-property optimization process, the mutual constraints between properties led to similar changing trends in the values of the three properties.

Fig. 2 and Figure E.5 in the Appendix show the property distributions of molecules generated with multi- and single-property optimization, respectively. Intuitively, in comparison to the molecule distributions in the training set (in blue), the chemical property distributions of the newly generated molecules (in green) shifted to the right overall. This suggests that InstGAN produced a greater number of new molecules with desirable properties. Furthermore, the property scores of molecules generated through multi-property optimization were marginally lower than those from single-property optimization.

This difference arises primarily because InstGAN had to concurrently consider the enhancement of three properties and several of which are inherently conflicting during multi-property optimization. Detailed results and trade-off analyses are provided in Appendix F. In essence, multi-property optimization for molecular generation proves to be more challenging and intricate.The results demonstrated the effectiveness of InstGAN in chemical property optimization.

4.3 Ablation Studies

Table 4 demonstrates the impact of various InstGAN variants on molecular generation performance. During the training process of InstGAN, we excluded GR, and MIE individually to create three distinct variants, namely, "w/o IR," "w/o GR," and "w/o MIE." These variants were then compared with InstGAN. In the "w/o IR" scenario, the calculation of instant reward was substituted with the MCTS-based RL algorithm. While MCTS enhanced GAN training stability with its extensive sampling, resulting in relatively high validity (97.80%), its computational complexity, stemming from the large number of samples, constrains its applicability in lengthy sequences. In the case of "w/o GR", the validity was the lowest (95.96%, compared to 97.56% of InstGAN). Given that the GR involves the global information of a SMILES string, it provides additional sequence-related information for generating subsequent



(a) Drug-likeness (QED).

(b) Solubility (logP).

(c) Synthesizability (SA).

Figure 2: Property distributions of generated molecules with multi-property optimization.

	Validity	Uniqueness	Novelty	Total
w/o IR	97.80	70.71	98.02	67.79
w/o GR	95.96	98.72	99.66	94.41
w/o MIE	98.39	96.50	99.54	94.51
InstGAN	97.56	98.47	99.73	95.81

Table 4: Effect of different variants of InstGAN.

tokens in the molecular auto-regression process, thereby contributing to the validity improvement. MIE, by smoothing the probabilities of generating tokens, enhances diversity in sampling tokens with non-maximum probabilities. Consequently, in the “w/o MIE” scenario, the generated molecular distribution exhibited the lowest uniqueness (96.50%, compared to 98.47% of InstGAN). InstGAN, incorporating IR, GR, and MIE, achieved the highest total score of 95.81%.

Additionally, Tables G.1, G.2, G.3, and G.4 display the effects of λ on the performance. InstGAN can be applied to extensive chemical databases, optimizing chemical properties while retaining a low computational cost.

Integrating an actor-critic RL with instant and global rewards is crucial to the success of InstGAN. When compared to MCTS sampling, our strategy is more computationally eifficient, and its fine-grained feedback over SMILES sub-strings provides informative gradients at the early training stage mitgating the gradient vanishing problem that often hinders discrete GANs. The same mechanism also stabilizes the learning process: in the integration setup of GAN and RL, the generator simultaneously minimizes KL divergence and maximizes JS divergence, producing conflicting gradients [Saxena and Cao, 2021]. Injecting entropy through token-level rewards and MIE can balance the two objectives and aligns with RL findings that entropy regularization facilitates exploration and robust learning [Ahmed *et al.*, 2019]. As a result, InstGAN achieves more stable convergence, improved sample quality, and performance on par with SOTA models.

4.4 Case Studies

In the case studies, our goal is to generate molecules with high QED and a significant affinity for DRD2 within the ChEMBL database. This pursuit is critical for identifying potential drug candidates distinguished by enhanced drug-like properties and targeted therapeutic effects. This significance is underscored by the wealth of available experimental bioactive data, providing a robust foundation for advancing promising compounds in drug discovery.

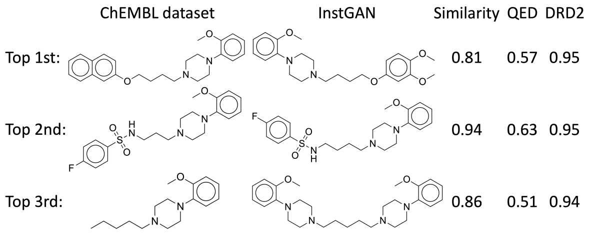


Figure 3: Comparison of the generated molecules with high DRD2 scores and similar approved drugs.

Table H.1 in Appendix H 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 3 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.

5 Conclusion

This study introduced InstGAN for generating molecules with multi-property optimization from scratch. Unlike MCTS-based RL algorithms, we employed an actor-critic RL algorithm for the efficient computation of IR and GR, resulting in reduced computation time and stabilized molecular generation quality. Additionally, the inclusion of MIE was used to alleviate the mode collapse problem and promote diversity in molecular generation. The experimental results demonstrated that InstGAN achieves comparable performance to SOTA baseline models and efficiently generates molecules with single- and multi-property optimization.

InstGAN has two main limitations. First, the number of critics increases with the number of chemical properties that need to be optimized, which leads to an increase in the training cost. Second, the inclusion of additional hyperparameters, such as λ and W_{C_i}, requires manual tuning, posing a challenge for fine-tuning. In future work, we will explore solutions to address these challenges.

Acknowledgements

This research was partially supported by the International Research Fellow of Japan Society for the Promotion of Science (Postdoctoral Fellowships for Research in Japan [Standard]) and KAKENHI (20K11830, 25K15130), Japan.

References

[Ahmed *et al.*, 2019] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

[Barshatski and Radinsky, 2021] Guy Barshatski and Kira Radinsky. Unpaired generative molecule-to-molecule translation for lead optimization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2554–2564, 2021.

[Barshatski *et al.*, 2021] Guy Barshatski, Galia Nordon, and Kira Radinsky. Multi-property molecular optimization using an integrated poly-cycle architecture. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3727–3736, 2021.

[Bickerton *et al.*, 2012] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.

[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.

[Chan *et al.*, 2019] HC Stephen Chan, Hanbin Shan, Thamani Dahoun, Horst Vogel, and Shuguang Yuan. Advancing drug discovery via artificial intelligence. *Trends in Pharmacological Sciences*, 40(8):592–604, 2019.

[Chen *et al.*, 2018] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug Discovery Today*, 23(6):1241–1250, 2018.

[Comer and Tam, 2001] John Comer and Kin Tam. Lipophilicity profiles: theory and measurement. *Pharmacokinetic Optimization in Drug Research: Biological, Physicochemical and Computational Strategies*, pages 275–304, 2001.

[Dai *et al.*, 2016] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *Proceedings of the International Conference on Machine Learning*, pages 2702–2711. PMLR, 2016.

[Dai *et al.*, 2018] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for molecule generation. In *Proceedings of the International Conference on Learning Representations*, 2018.

[De Cao and Kipf, 2018] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv Preprint arXiv:1805.11973*, 2018.

[de Masson d'Autume *et al.*, 2019] Cyprien de Masson d'Autume, Shakir Mohamed, Mihaela Rosca, and Jack Rae. Training language GANs from scratch. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 32, 2019.

[Dollar *et al.*, 2021] Orion Dollar, Nisarg Joshi, David AC Beck, and Jim Pfaendtner. Attention-based generative models for de novo molecular design. *Chemical Science*, 12(24):8362–8372, 2021.

[Ertl and Schuffenhauer, 2009] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):1–11, 2009.

[Fedus *et al.*, 2018] William Fedus, Ian Goodfellow, and Andrew M Dai. MaskGAN: better text generation via filling in the_. *arXiv Preprint arXiv:1801.07736*, 2018.

[Gaulton *et al.*, 2017] Anna Gaulton, Anne Hersey, Michał Nowotka, A Patricia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J Bellis, Elena Cibrián-Uhalte, et al. The chembl database in 2017. *Nucleic Acids Research*, 45(D1):D945–D954, 2017.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 27, 2014.

[Guimaraes *et al.*, 2017] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv Preprint arXiv:1705.10843*, 2017.

[Guo *et al.*, 2023] Siyuan Guo, Jihong Guan, and Shuigeng Zhou. Diffusing on two levels and optimizing for multiple properties: A novel approach to generating molecules with desirable properties. *arXiv Preprint arXiv:2310.04463*, 2023.

[Irwin *et al.*, 2012] John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.

[Jin *et al.*, 2018] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *Proceedings of the International Conference on Machine Learning*, pages 2323–2332. PMLR, 2018.

[Jin *et al.*, 2019] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical graph-to-graph translation for molecules. *arXiv Preprint arXiv:1907.11223*, 2019.

[Jo *et al.*, 2022] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proceedings of the International Conference on Machine Learning*, pages 10362–10383. PMLR, 2022.

[Kingma and Dhariwal, 2018] Durk P Kingma and Prafulla Dhariwal. GLow: Generative flow with invertible 1x1 convolutions. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 31, 2018.

[Klein and Trinajstic, 1984] DJ Klein and N Trinajstic. Hückel rules and electron correlation. *Journal of the American Chemical Society*, 106(26):8050–8056, 1984.

[Konda and Tsitsiklis, 1999] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 12, 1999.

[Kusner *et al.*, 2017] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the International Conference on Machine Learning*, pages 1945–1954. PMLR, 2017.

[Landrum, 2013] Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.

[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 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.

[Lippe and Gavves, 2021] Phillip Lippe and Efstratios Gavves. Categorical normalizing flows via continuous transformations. In *Proceedings of the International Conference on Learning Representations*, 2021.

[Luo *et al.*, 2021] Youzhi Luo, Keqiang Yan, and Shuiwang Ji. GraphDF: A discrete flow model for molecular graph generation. In *Proceedings of the International Conference on Machine Learning*, pages 7192–7203. PMLR, 2021.

[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.

[OpenAI, 2023] R OpenAI. GPT-4 technical report. arxiv 2303.08774. *View in Article*, 2023.

[Paul *et al.*, 2021] Debleena Paul, Gaurav Sanap, Snehal Shenoy, Dnyaneshwar Kalyane, Kiran Kalia, and Rakesh K Tekade. Artificial intelligence in drug discovery and development. *Drug Discovery Today*, 26(1):80, 2021.

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

[Saxena and Cao, 2021] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54(3):1–42, 2021.

[Shi *et al.*, 2020] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: a flow-based autoregressive model for molecular graph generation. *arXiv Preprint arXiv:2001.09382*, 2020.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Tang *et al.*, 2023] Huidong Tang, Chen Li, Shuai Jiang, Huachong Yu, Sayaka Kamei, Yoshihiro Yamanishi, and Yasuhiko Morimoto. Earlgan: An enhanced actor–critic reinforcement learning agent-driven gan for de novo drug design. *Pattern Recognition Letters*, 175:45–51, 2023.

[Vignac *et al.*, 2023] Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

[Weininger, 1988] David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[Yu *et al.*, 2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[Zang and Wang, 2020] Chengxi Zang and Fei Wang. MoFlow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 617–626, 2020.

[Zhang *et al.*, 2021] Shifeng Zhang, Ning Kang, Tom Ryder, and Zhenguo Li. Iflow: Numerically invertible flows for efficient lossless compression via a uniform coder. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 34, pages 5822–5833, 2021.