ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447350

A PROPERTY-GUIDED DIFFUSION MODEL FOR GENERATING MOLECULAR GRAPHS

Changsheng Ma[⋆], *Taicheng Guo*[†], *Qiang Yang*[⋆], *Xiuying Chen*[⋆], *Xin Gao*[⋆]
Shangsong Liang[*], *Nitesh Chawla*[†], *Xiangliang Zhang*[†]

[⋆] King Abdullah University of Science and Technology, Jeddah, Saudi Arabia
[*] MBZUAI, UAE [†] University of Notre Dame, IN, US

ABSTRACT

Inverse molecular generation is an essential task for drug discovery, and generative models offer a very promising avenue, especially when diffusion models are used. Despite their great success, existing methods are inherently limited by the lack of a semantic latent space that can not be navigated and perform targeted exploration to generate molecules with desired properties. Here, we present a property-guided diffusion model for generating desired molecules, which incorporates a sophisticated diffusion process capturing intricate interactions of nodes and edges within molecular graphs and leverages a time-dependent molecular property classifier to integrate desired properties into the diffusion sampling process. Furthermore, we extend our model to a multi-property-guided paradigm. Experimental results underscore the competitiveness of our approach in molecular generation, highlighting its superiority in generating desired molecules without the need for additional optimization steps.

***Index Terms*—** Molecular Graph Generation, Diffusion Model, Drug Discovery

1. INTRODUCTION

A fundamental problem in drug discovery and chemistry is to design novel molecules with desired properties. Direct optimization or exploration of the vast and discrete space of drug-like molecules, estimated to be on the order of 10^{60} [1], is daunting. Recent advances in deep generative models have led to significant progress in the field, especially after the introduction of diffusion models [2], which significantly enhanced the capacity to capture the underlying data distribution and generate valid molecules. However, their inability to generate molecules with specific properties is a fatal limitation. This limitation arises from the standard two-stage process in generating desired molecules: encoding the original dataset into a semantic latent space, followed by conducting Bayesian optimization within this space to do optimization. Since the latent space of the diffusion model lacks semantics, interpolation in this latent space along a specific direction leads to unpredictable changes and fails to produce the desired molecules, as shown in Fig. 1.

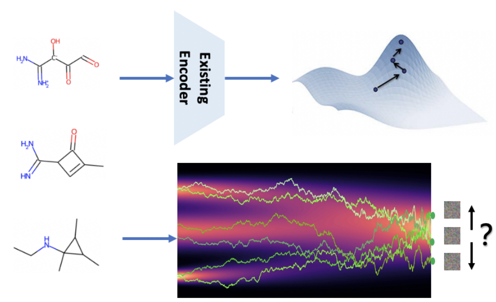

Fig. 1: The latent space of existing generators (upper) possesses semantics, allowing for Bayesian optimization to find desired molecules, whereas the diffusion model cannot, because its latent space (lower) has no semantics.

In this paper, we exploit a classifier to improve the diffusion generator and introduce a property-guided diffusion model for molecular graph generation. Unlike the previous two-stage paradigm, our approach leverages the synergy between a graph diffusion generator and a time-dependent molecular property classifier. By training the property classifier with respect to specific molecular property categories, we harness the gradient information derived from the classifier over time steps to guide the diffusion sampling process toward the desired property category. This succinct design comes with two additional advantages, one is to generate the desired molecule without an additional optimization process, and the other is that molecules with multi-objective properties can be generated by simply linearly combining multiple property classifiers. We validate our method on molecule generation and desired molecule generation, the results show that our model can generate the desired molecules well while maintaining competitive generation performance.

Our contributions are as follows: 1) To the best of our knowledge, we are the first to propose the property-guided diffusion model for molecular graph generation, providing an effective solution to generate desired molecules via diffusion models. 2) For more practical use, we extend our model to multi-property-guided molecular generation, enabling the concurrent satisfaction of multiple properties. 3) Experimental results demonstrate the superior performance of our method over state-of-the-art baselines.

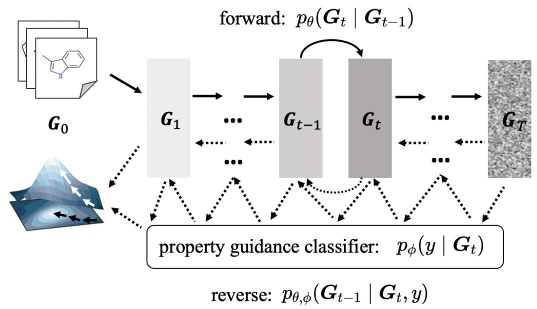

Fig. 2: The overview of our model: the solid arrow represents the forward process, and the dashed arrows depict the property-guided reverse process.

2. METHODOLOGY

In this section, we start with problem formulation and then provide the theoretical foundation of the proposed model for building a graph diffusion model with classifier guidance as shown in Fig. 2. Subsequently, we delve into the specifics of the graph diffusion generator and the property guidance classifier. Lastly, we outline the sampling process.

2.1. Problem Formulation

Let $\boldsymbol{G} = (\boldsymbol{A}, \boldsymbol{X})$ denote a molecular graph $\boldsymbol{G}$ with adjacency tensor $\boldsymbol{A}$ and feature matrix $\boldsymbol{X}$. To model the diffusion process, we introduce timesteps denoted as T. We initialize the process with $\boldsymbol{G}_0$ and define a noisy trajectory as $\{\boldsymbol{G}_t = (\boldsymbol{X}_t, \boldsymbol{A}_t)\}_{t\in[0,T]}$, where $[0, T]$ denotes a fixed timestep range. Our aim is to generate novel molecules while maximizing alignment with a specific property category y.

2.2. Theoretical Foundation

The forward process of diffusion models defines a Markov chain in which random noise is progressively added to the data until the output distribution converges to a known prior distribution, such as a Gaussian distribution. In the context of the graph domain, we begin by randomly sampling a graph $\boldsymbol{G}_0$, and the well-defined forward process unfolds over T timesteps, resulting in a trajectory of graphs $\{\boldsymbol{G}_t\}_{t\in[0,T]}$, which satisfy the following equation:

$$q(\boldsymbol{G}_t \mid \boldsymbol{G}_0) = \sqrt{\overline{\alpha}_t}\boldsymbol{G}_0 + \epsilon\sqrt{1-\overline{\alpha}_t}, \epsilon \sim \mathcal{N}(0,1), \tag{1}$$

where $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and α_t is a time-dependant differentiable function chosen to ensure $q(\boldsymbol{G}_t) \approx \mathcal{N}(\boldsymbol{G}_t; \mathbf{0}, \mathbf{I})$. We model this diffusion process by Stochastic Differential Equations (SDEs) and obtain the graph diffusion generator p_θ. Then the reverse noising process follows $p_\theta(\boldsymbol{G}_{t-1} \mid \boldsymbol{G}_t) = \mathcal{N}\left(\mu_{\theta_t}, \sigma_{\theta_t}^2\mathbf{I}\right)$, where μ_{θ_t} and $\sigma_{\theta_t}^2$ are the mean and variance under the reverse process at time t.

To generate a graph that satisfies a specific category y, we train a property guidance classifier $p_\phi(y \mid \boldsymbol{G}_t)$ on noisy graphs $\boldsymbol{G}_t$ to guide the diffusion sampling process. Expanding on [3], with Z as a normalizing constant, the distribution post-incorporating the classifier guidance is as follows:

$$p_{\theta,\phi}(\boldsymbol{G}_{t-1} \mid \boldsymbol{G}_t, y) = Zp_\theta(\boldsymbol{G}_{t-1} \mid \boldsymbol{G}_t)\, p_\phi(y \mid \boldsymbol{G}_{t-1}). \tag{2}$$

This distribution can be approximated by a Gaussian distribution with a shifted mean as:

$$p_\theta(\boldsymbol{G}_{t-1} \mid \boldsymbol{G}_t)\, p_\phi(y \mid \boldsymbol{G}_{t-1}) = \mathcal{N}(\mu_{\theta_t} + \sigma_{\theta_t}^2\mathbf{I}g, \sigma_{\theta_t}^2), \tag{3}$$

where $p_\phi(y \mid \boldsymbol{G}_{t-1})$ is the probability of $\boldsymbol{G}_{t-1}$ belonging to class y and $g = \nabla_{\boldsymbol{G}_{t-1}} \log p_\phi(y \mid \boldsymbol{G}_{t-1})$.

2.3. Graph Diffusion Generator

Given a graph $\boldsymbol{G}_0$ and its trajectory of noisy random variables $\{\boldsymbol{G}_t = (\boldsymbol{X}_t, \boldsymbol{A}_t)\}_{t\in[0,T]}$ over a fixed range of timesteps $[0, T]$, the forward process can be represented by SDEs as

$$\mathrm{d}\boldsymbol{G}_t = \mathbf{f}_t(\boldsymbol{G}_t)\,\mathrm{d}t + \mathbf{g}_t(\boldsymbol{G}_t)\,\mathrm{d}\mathbf{w}, \tag{4}$$

where $\mathbf{f}_t(\cdot)$ represents the linear drift coefficient and $\mathbf{g}_t(\cdot)$ denotes the diffusion coefficient. Here, $\mathbf{w}$ represents the standard Wiener process. For the sake of simplicity, we choose $g_t(\boldsymbol{G}_t)$ to be a scalar function g_t.

The generation process, which corresponds to the reverse process of Eq. (4), can also be modeled as a diffusion process, as demonstrated by [4], following the reverse-time SDEs as

$$\mathrm{d}\boldsymbol{G}_t = \left[\mathbf{f}_t(\boldsymbol{G}_t) - g_t^2\nabla_{\boldsymbol{G}_t}\log p_t(\boldsymbol{G}_t)\right]\mathrm{d}\bar{t} + g_t\,\mathrm{d}\overline{\mathbf{w}}, \tag{5}$$

where $\nabla_{\boldsymbol{G}_t}\log p_t(\boldsymbol{G}_t)$ represents the graph score function, $\overline{\mathbf{w}}$ denotes a reverse-time standard Wiener process, and $\mathrm{d}t$ represents infinitesimal negative timesteps from T to 0. For the convenience of computation, we further decompose the reverse-time SDEs into two components, namely the nodes component and the adjacency matrix component:

$$\begin{aligned} \mathrm{d}\boldsymbol{X}_t &= \left[\mathbf{f}_{1,t}(\boldsymbol{X}_t) - g_{1,t}^2\nabla_{\boldsymbol{X}_t}\log p_t(\boldsymbol{G}_t)\right]\mathrm{d}\bar{t} + g_{1,t}\,\mathrm{d}\overline{\mathbf{w}}_1, \\ \mathrm{d}\boldsymbol{A}_t &= \left[\mathbf{f}_{2,t}(\boldsymbol{A}_t) - g_{2,t}^2\nabla_{\boldsymbol{A}_t}\log p_t(\boldsymbol{G}_t)\right]\mathrm{d}\bar{t} + g_{2,t}\,\mathrm{d}\overline{\mathbf{w}}_2, \end{aligned} \tag{6}$$

where $\mathbf{f}_{1,t}$ and $\mathbf{f}_{2,t}$ are linear drift coefficients, satisfying $\mathbf{f}_t(\boldsymbol{X}, \boldsymbol{A}) = (\mathbf{f}_{1,t}(\boldsymbol{X}), \mathbf{f}_{2,t}(\boldsymbol{A}))$. Similarly, $g_{1,t}$ and $g_{2,t}$ are scalar diffusion coefficients, and $\overline{\mathbf{w}}_1, \overline{\mathbf{w}}_2$ denote reverse-time standard Wiener processes. Consequently, the problem of graph diffusion now translates into diffusing the nodes and adjacency matrix while preserving their correlation over time.

Therefore, our graph diffusion generator p_θ comprises two neural networks, namely $\boldsymbol{s}_{\gamma,t}$ and $\boldsymbol{s}_{\delta,t}$, responsible for estimating the two partial score functions $\nabla_{\boldsymbol{X}_t}\log p_t(\boldsymbol{G}_t)$ and $\nabla_{\boldsymbol{A}_t}\log p_t(\boldsymbol{G}_t)$, respectively. To achieve this, we minimize the following objective functions:

$$\begin{aligned} &\min_{\gamma}\mathbb{E}_t\left\{\lambda_1(t)\mathbb{E}_{\boldsymbol{G}_0}\mathbb{E}_{\boldsymbol{G}_t\mid\boldsymbol{G}_0}\left\|\boldsymbol{s}_{\gamma,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{X}_t}\log p_t(\boldsymbol{G}_t)\right\|_2^2\right\}, \\ &\min_{\delta}\mathbb{E}_t\left\{\lambda_2(t)\mathbb{E}_{\boldsymbol{G}_0}\mathbb{E}_{\boldsymbol{G}_t\mid\boldsymbol{G}_0}\left\|\boldsymbol{s}_{\delta,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{A}_t}\log p_t(\boldsymbol{G}_t)\right\|_2^2\right\}, \end{aligned} \tag{7}$$

where $\lambda_1(t)$ and $\lambda_2(t)$ are positive weighting functions. However, the ground-truth partial scores are not analytically tractable. Therefore, we employ the denoising score matching method proposed by [5] to estimate the partial scores.

The idea is to replace $p_t(\boldsymbol{G}_t)$ by $p_{0t}(\boldsymbol{G}_t \mid \boldsymbol{G}_0)$, which is the transition distribution from p_0 to p_t induced by the forward diffusion process. Importantly, due to the linearity of the drift coefficient, this transition distribution can be expressed as $p_{0t}(\boldsymbol{G}_t \mid \boldsymbol{G}_0) = p_{0t}(\boldsymbol{X}_t \mid \boldsymbol{X}_0)\, p_{0t}(\boldsymbol{A}_t \mid \boldsymbol{A}_0)$. Sampling from these Gaussian distributions is straightforward since the means and variances are tractable and determined by the coefficients of the forward diffusion process. Thus, the objective functions can be expressed as follows:

$$\min_{\gamma} \mathbb{E}_t \left\{ \lambda_1(t) \mathbb{E}_{\boldsymbol{G}_0} \mathbb{E}_{\boldsymbol{G}_t|\boldsymbol{G}_0} \left\| \boldsymbol{s}_{\gamma,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{X}_t} \log p_{0t}(\boldsymbol{X}_t \mid \boldsymbol{X}_0) \right\|_2^2 \right\},$$
$$\min_{\delta} \mathbb{E}_t \left\{ \lambda_2(t) \mathbb{E}_{\boldsymbol{G}_0} \mathbb{E}_{\boldsymbol{G}_t|\boldsymbol{G}_0} \left\| \boldsymbol{s}_{\delta,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{A}_t} \log p_{0t}(\boldsymbol{A}_t \mid \boldsymbol{A}_0) \right\|_2^2 \right\}. \tag{8}$$

To capture the dependency between $\boldsymbol{X}_t$ and $\boldsymbol{A}_t$ and improve the estimation of the partial score functions, we employ different neural network architectures. For the score-based model $\boldsymbol{s}_{\gamma,t}$, which estimates $\nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{G}_t)$, we utilize multiple layers of Graph Convolutional Network (GCN) [6]. For the score-based model $\boldsymbol{s}_{\delta,t}$, which is responsible for estimating $\nabla_{\boldsymbol{A}_t} \log p_t(\boldsymbol{G}_t)$, it is crucial to consider the structural dependencies between nodes due to the significant impact of bonds on the chemical structure of a molecule. Thus, we leverage graph multi-head attention [7] to capture node interactions based on their structural dependencies. Additionally, we utilize the two-order adjacency matrix to model long-range dependencies within the molecule, further enhancing the ability of the model to capture the intricate relationship between nodes in the molecular graph.

Furthermore, we incorporate the time information into the two score-based models $\boldsymbol{s}_{\gamma,t}$ and $\boldsymbol{s}_{\delta,t}$, by scaling the output of the models with the standard deviation of the transition distribution at time t. This scaling allows the models to adjust their predictions based on the uncertainty of the diffusion process at each timestep. By considering the time-dependent scaling, we can better align the estimated partial score functions with the properties of the underlying graph at different stages of the diffusion process.

2.4. Propertiy Guidance Classifier

Considering that molecular properties are global features, we leverage GCN [6] along with the gated recurrent unit from [8] to develop our classifier p_ϕ. Importantly, the models utilized for the reverse process and guidance in Equation (3) are time-dependent, operating on noisy input graphs. Therefore, the property guidance classifier p_ϕ must consider the timestep t as an additional input and undergo training on noisy graphs from various timesteps. The objective function is the entropy loss between the predicted property category and the desired property category as $\mathcal{L}_{p_\phi} = \mathrm{CrossEntropy}\left(p_\phi(\boldsymbol{G}_t), y\right)$.

By integrating the gradient $\nabla_{\boldsymbol{G}_t} \log p_\phi(\boldsymbol{G}_t, y)$ into the sampling process, we can ensure the generated molecules exhibit the desired characteristics or meet certain criteria defined by the classifier. Moreover, for generating desired molecules with multiple properties, our model can seamlessly extend to multi-property guidance by integrating a weighted sum of each classifier as $p_\phi(y \mid \boldsymbol{G}_t) = \sum_i \rho_i p_{\phi_i}(y_i \mid \boldsymbol{G}_t)$, where ρ_i is the weighting factor and $p_{\phi_i}(y_i \mid \boldsymbol{G}_t)$ denotes the classifier for each property.

Algorithm 1 Property-Guided Diffusion Sampling

Input: Given a diffusion model p_θ, classifier p_ϕ, and property category y, gradient scales for $\boldsymbol{X}$ and $\boldsymbol{A}$: $\rho_{\boldsymbol{X}}, \rho_{\boldsymbol{A}}$
Output: Generated nodes feature $\boldsymbol{X}_0$, adjacency matrix $\boldsymbol{A}_0$

1: $\boldsymbol{X}_T, \boldsymbol{A}_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
2: **for** $t = T$ to 1 **do**
3: $\quad \mu_{\boldsymbol{X}_t}, \Sigma_{\boldsymbol{X}_t}, \mu_{\boldsymbol{A}_t}, \Sigma_{\boldsymbol{A}_t} \leftarrow p_{\theta_t}$
4: $\quad \boldsymbol{X}_{t-1} \leftarrow$ sample from $\mathcal{N}\left(\mu_{\boldsymbol{X}_t} + \rho_{\boldsymbol{X}} \Sigma_{\boldsymbol{X}_t} \nabla_{\boldsymbol{X}_t} \log p_{\phi_t}(\boldsymbol{G}_t, y), \Sigma_{\boldsymbol{X}_t}\right)$
5: $\quad \boldsymbol{A}_{t-1} \leftarrow$ sample from $\mathcal{N}\left(\mu_{\boldsymbol{A}_t} + \rho_{\boldsymbol{A}} \Sigma_{\boldsymbol{A}_t} \nabla_{\boldsymbol{A}_t} \log p_{\phi_t}(\boldsymbol{G}_t, y), \Sigma_{\boldsymbol{A}_t}\right)$
6: **end for**
7: **return** $\boldsymbol{X}_0, \boldsymbol{A}_0$

2.5. Graph Sampling Process

To generate graphs, we simulate reverse-time SDE trajectories as in Equation (6) while incorporating property classifier guidance at each timestep. We adopt the Predictor-Corrector Sampler [5], as this strategy enables effective exploration of high-density distribution regions while avoiding low-density ones. The predictor relies on the reverse diffusion SDE solver (the SDE employed in our approach is VE SDE [5]), and the corrector employs annealed Langevin dynamics [9]. Once we have obtained the graph diffusion model p_θ and the classifier model p_ϕ, we utilize the gradients $\nabla_{\boldsymbol{X}_t} \log p_\phi(\boldsymbol{G}_t, y)$ and $\nabla_{\boldsymbol{A}_t} \log p_\phi(\boldsymbol{G}_t, y)$ to guide the diffusion sampling process towards the property category y. The overall sampling process is summarized in Algorithm 1.

3. EXPERIMENT

3.1. Experimental Setup

Dataset. We experiment on QM9 [10], utilizing plogP [11] and QED [12] as guidance properties. The datasets are divided into four categories based on property value quartiles.
Baselines. We compare our model with several SOTA models: GraphAF [13], GraphDF [14] and MoFlow [15] are flow-based models. GraphEBM [16] is an energy-based model. EDP-GNN [17] and GDSS [18] are diffusion-based models.
Evaluation Metrics. ***Validity*** is the percentage of chemically valid molecules, ***Uniqueness*** is the percentage of unique molecules, and ***Novelty*** is the percentage of novel molecules with reference to the training set. ***V.U.N*** is the production of these three metrics. ***Validity w/o Correction (VwoC)*** is the percentage of valid molecules without post-hoc chemical valency correction. ***Fréchet ChemNet Distance (FCD)*** [19] is to evaluate the distance between the training and generated

Table 1: **Generation performance on QM9. Results are the means and standard deviations of three independent runs.**

Method	% VwoC ↑	NSPDK ↓	FCD ↓	% Validity ↑	% Uniqueness ↑	% Novelty ↑	% V.U.N ↑
GraphAF	67	0.020 ± 0.003	5.268 ± 0.403	**100.00**	94.51	88.83	83.95
GraphDF	82.67	0.063 ± 0.001	10.816 ± 0.020	**100.00**	97.62	**98.10**	95.77
MoFlow	91.36 ± 1.23	0.017 ± 0.003	4.467 ± 0.595	$\mathbf{100.00 \pm 0.00}$	98.65 ± 0.57	94.72 ± 0.77	93.44 ± 0.44
EDP-GNN	47.52 ± 3.60	0.005 ± 0.001	2.680 ± 0.221	$\mathbf{100.00 \pm 0.00}$	$\mathbf{99.25 \pm 0.05}$	86.58 ± 1.85	85.93 ± 0.09
GraphEBM	8.22 ± 2.24	0.030 ± 0.004	6.143 ± 0.411	$\mathbf{100.00 \pm 0.00}$	97.90 ± 0.14	97.01 ± 0.17	94.97 ± 0.02
GDSS	95.72 ± 1.94	0.003 ± 0.000	2.900 ± 0.282	$\mathbf{100.00 \pm 0.00}$	98.46 ± 0.61	86.27 ± 2.29	84.94 ± 1.40
Ours	$\mathbf{96.98 \pm 1.23}$	$\mathbf{0.002 \pm 0.000}$	$\mathbf{2.204 \pm 0.065}$	$\mathbf{100.00 \pm 0.00}$	98.52 ± 0.15	97.23 ± 1.05	$\mathbf{95.79 \pm 0.16}$

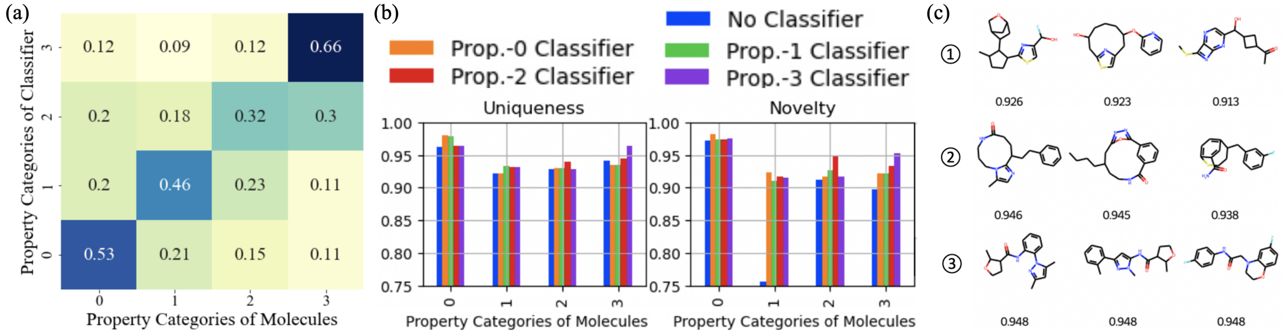

Fig. 3: Property-Guided performance. (a): The proportion of molecules in each category relative to all molecules generated under the guidance of different classifiers. (b): The uniqueness and novelty value of generated molecules within each individual class under the guidance of different classifiers. (c): Top 3 molecules with QED values out of 10,000 randomly generated molecules by different models: ① Without guidance, ② QED-guided, ③ QED & Ring-guided.

graph sets. ***Neighborhood subgraph pairwise distance kernel (NSPDK)*** [20] calculates the mean maximum discrepancy between the generated and test molecules.

Implementation Details. For the property classifier p_ϕ, we utilize a GCN with 2 layers, with hidden dimensions set to 16. The scaling factors $\rho_{\boldsymbol{X}}$ and $\rho_{\boldsymbol{A}}$ are set to 0.6 and 0.0, respectively. For the diffusion model p_θ, $\boldsymbol{s}_\gamma$ comprises a GCN with 2 layers and a hidden dimension of 16. $\boldsymbol{s}_\delta$ employs an attention-based architecture with 4 attention heads, initial, hidden, and final channel sizes of 2, 8, and 4, respectively. The number of GCN layers is 3 with a hidden dimension of 16. The SDEs process follows [18]. During training, we use the Adam optimizer with a learning rate of 5×10^{-3} and weight decay of 1×10^{-4}. The batch size is 1024. We perform 300 training epochs for p_θ and 50 epochs for p_ϕ.

3.2. Results and Discussion

Generation Performance. All metrics are assessed on 10,000 randomly generated molecules, with our model outperforming baselines in almost all metrics. The highest VwoC value highlights the ability of our model to grasp the chemical valency rule, while the top NSPDK and FCD values indicate its capability to capture the underlying distribution. Furthermore, the best V.U.N value shows our model can generate more valid, unique, and novel molecules.

Property-Guided Performance. As the property category is defined through quartile divisions of the property value of the original dataset, we perform the sampling process guided by each property classifier and conduct inter-class evaluation and intra-class evaluation. Fig. 3(a) shows that different property classifiers can guide the generation of a greater number of molecules within their respective categories, providing satisfactory results in inter-class evaluation. Fig. 3(b) shows molecules generated under corresponding property guidance within each class exhibit improved uniqueness and novelty, demonstrating high quality in intra-class evaluation. Overall, the property-guided performance of our model is verified.

Multi-Property-Guided Performance. To align more closely with drug design scenarios, we extend our model to handle larger molecules and multiple properties. We introduce Ring-guided diffusion alongside the QED-guided diffusion, setting the guidance for the ring number to three. This choice is motivated by the empirical observation that molecules with high QED values typically exhibit a characteristic of three rings. In Fig. 3(c), ② exhibits better QED values compared to ①, and ③ outperforms both, highlighting the effectiveness of our model in multi-property-guided scenarios.

4. CONCLUSION

In this paper, we present a novel property-guided diffusion model for molecular graph generation, where a time-dependent classifier is integrated to guide the diffusion sampling process toward desired property categories. This addresses the limitation of conventional diffusion models, which lack a semantic latent space for targeted molecular optimization. Experimental results demonstrate the superiority of our model in generating molecules with desired properties.

5. REFERENCES

[1] Asher Mullard, "The drug-maker's guide to the galaxy," *Nature News*, vol. 549, no. 7673, pp. 445, 2017.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[3] Prafulla Dhariwal and Alexander Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[4] Brian DO Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[5] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[6] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[7] Jinheon Baek, Minki Kang, and Sung Ju Hwang, "Accurate learning of graph representations with graph multiset pooling," *arXiv preprint arXiv:2102.11533*, 2021.

[8] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[9] Yang Song and Stefano Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[10] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific data*, vol. 1, no. 1, pp. 1–7, 2014.

[11] Peter Ertl and Ansgar Schuffenhauer, "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions," *Journal of cheminformatics*, vol. 1, no. 1, pp. 1–11, 2009.

[12] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins, "Quantifying the chemical beauty of drugs," *Nature chemistry*, vol. 4, no. 2, pp. 90–98, 2012.

[13] 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.

[14] Youzhi Luo, Keqiang Yan, and Shuiwang Ji, "Graphdf: A discrete flow model for molecular graph generation," *arXiv preprint arXiv:2102.01189*, 2021.

[15] Chengxi Zang and Fei Wang, "Moflow: an invertible flow model for generating molecular graphs," in *SIGKDD*, 2020, pp. 617–626.

[16] Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji, "Graphebm: Molecular graph generation with energy-based models," *arXiv preprint arXiv:2102.00546*, 2021.

[17] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon, "Permutation invariant graph generation via score-based generative modeling," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 4474–4484.

[18] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang, "Score-based generative modeling of graphs via the system of stochastic differential equations," *arXiv preprint arXiv:2202.02514*, 2022.

[19] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer, "Fréchet chemnet distance: a metric for generative models for molecules in drug discovery," *Journal of chemical information and modeling*, vol. 58, no. 9, pp. 1736–1741, 2018.

[20] Fabrizio Costa and Kurt De Grave, "Fast neighborhood subgraph pairwise distance kernel," in *Proceedings of the 26th International Conference on Machine Learning*. Omnipress; Madison, WI, USA, 2010, pp. 255–262.