MoE-Guided Graph Diffusion for Oriented Molecule Design

Shuochen Li[1], **Xiangqi Guo**[2], **Huobin Tan**[1*], **Lei Shi**[1*]

[1]Beihang University, Beijing, China
[2]Peking University, Beijing, China
lisc07@buaa.edu.cn, 2501210249@stu.pku.edu.cn, thbin@buaa.edu.cn, leishi@buaa.edu.cn

Abstract

Designing molecules with desired properties, aka the oRiented molEcule Design (RED), is a fundamental task in chemistry and materials science. While graph diffusion models (GDMs) and reinforcement learning techniques (RL) show promise in molecule structure generation and property optimization stages individually, their integration in the unified RED task often suffers from poor compatibility. The large variance among candidate molecular structures generated by GDMs can be amplified in the iterative optimization process of RL, leading to slow and unstable convergence. In this work, motivated by the adaptive and divide-and-conquer characteristics of Mixture of Experts (MoE) architecture, we propose a novel framework called MoE-Guided Graph Diffusion Model (MEGD) that incorporates the MoE architecture to guide the orchestration of GDM and RL, promoting faster and more stable convergence in the design process. MEGD is evaluated on benchmark datasets optimizing the physical and chemical properties of AI-generated molecular structures. On all three datasets, our method outperforms the best of 9 alternative models by 7.73% on the target structural properties, while not penalizing other important application-level quality metrics of the generated molecules. A real-world case study on an emerging class of material, i.e., metal-organic framework, is also conducted, which further demonstrates the effectiveness of our method in accomplishing the RED task.

Introduction

Designing novel molecules with desired properties, aka the oRiented molEcule Design (RED) problem, is an emerging challenge in chemistry and materials science, with important applications for drug discovery, energy storage, and advanced material production (Pang et al. 2025). For example, synthesizing metal-organic frameworks (MOFs) with large specific surface area (SSA) can dramatically increase the capability of capturing CO_2 for greenhouse gas reduction (Khan et al. 2024). Traditional trial-and-error methods for molecular design are often time-consuming or even intractable, primarily due to the vastness of their design space, which is estimated to include up to 10^{60} feasible molecules (Virshup et al. 2013). Recently, artificial intelligence (AI) methods, such as deep generative models (Jing et al. 2024; Du et al. 2024), have gained increasing attention as they are more efficient in exploring the huge design space and accelerating the molecule discovery process.

In this work, we focus on designing the graph-based molecule structure (see Preliminaries for formal definition), as they largely determine the physical and chemical properties of their top-level material/chemistry class. In the AI4Science community, diffusion models have long been borrowed from the computer vision field (Cao et al. 2024) to generate graph and topology structure mimicking their real-world counterpart in natural science, which is generally called graph diffusion models (GDM). Meanwhile, to attain the desired property of a molecule, such as the SSA of a MOF, the default GDM should be extended to support property-oriented graph structure design. Among various alternatives, reinforcement learning techniques (RL) have shown promising records in optimizing specific objective function of learning-based systems, which is a straightforward method to extend GDM.

The main problem we solve in this paper is that GDM and RL have poor compatibility when combined. GDM exhibits large variance in sampling candidate graph structures along the diffusion process (Yang et al. 2023). This variance is further amplified by the RL policy estimation step (Romoff et al. 2018), which finally leads to slow or even failed convergence in the closed-loop oriented molecule design process. In this work, we are motivated by the excellent performance achieved when introducing the Mixture of Experts (MoE) architecture to state-of-the-art LLMs (Dai et al. 2024). Analogously, we propose a new GDM framework called MoE-Guided Graph Diffusion Models (MEGD), which integrates the MoE layers into both GDM architecture and RL mechanism for solving the RED problem. The divide-and-conquer strategy of MoE greatly reduces the variance of individual gradient estimates, while the new adaptive gating network dynamically routes RL reward to the most relevant expert for policy gradient update, leading to faster and more stable convergence in molecule structure optimization. In more detail, the MEGD framework consists of two stages, first pretraining an optimized GDM with MoE to preserve generic molecule structure generation capability, and then finetuning the model with RL for maximizing the desired molecule graph property, ultimately accomplishing the oriented molecule design task.

*Corresponding author:{thbin, leishi}@buaa.edu.cn

The main contribution of this work can be summarized as:

- **Efficient generative model for oriented molecule design**: the proposed framework incorporates an MoE architecture to resolve the compatibility problem when combining GDMs and RL techniques, and achieves accelerated convergence in the closed-loop design process;
- **Optimized MoE mechanism for graph diffusion models**: by introducing gated routing and expert network structure, we assign molecule graphs of different time steps and substructures to separate and most relevant experts for processing, making the structure generation process more stable and effective;
- **Demonstrated application to real-world chemical and material design**: we apply the proposed framework to the actual design of MOF polymers and achieve promising results on benchmark tests.

Related Work

Graph diffusion models for molecule generation

Early graph generation solutions mainly rely on auto-regressive models (You et al. 2018b; Popova et al. 2019), variational autoencoders (Liu et al. 2018; Jin, Barzilay, and Jaakkola 2018), normalizing flows (Madhawa et al. 2019; Luo, Yan, and Ji 2021), and generative adversarial networks (De Cao and Kipf 2018; Maziarka et al. 2020). While effective for specific graph classes, they struggle with general graph generation due to the challenge of capturing permutation-invariant properties.

Graph diffusion models address the limitation of previous methods by offering a more flexible and symmetry-aware framework for graph generation. Among many proposals, EDP-GNN (Niu et al. 2020) was one of the first to apply diffusion processes to graph generation, followed by well-known models such as GDSS (Jo, Lee, and Hwang 2022), GraphGDP (Huang et al. 2022), DiGress (Vignac et al. 2022), and GCDM (Morehead and Cheng 2024). These models enhance the flexibility, validity, and permutation-invariance of the graph generation process.

Meanwhile, the adaptation of graph diffusion models to the molecule graph structure generation task brings additional challenges. The generated molecule graphs should preserve chemical fidelity, including validity and stability, in addition to satisfying generic graph structure constraints such as permutation and E(3) invariance. To address these specific challenges, recent studies have customized graph diffusion models for molecule generation from various aspects. GeoDiff (Xu et al. 2022) employs a Graph Field Network (GFN) to model each atom as a particle, enabling the generation of stable molecule conformations. MolDiff (Peng et al. 2023) introduces an E(3)-equivariant framework that jointly diffuses atoms and bonds to resolve their inconsistencies. PMDM (Huang et al. 2024) incorporates a dual-equivariant diffusion mechanism into the 3D molecule modeling framework, therefore overcomes the inherent limitation of auto-regressive methods.

Existing studies on GDM-based molecule graph generation pioneer the transition from general graph generation to chemically valid molecule design. However, on the RED task with coinciding graph generation and property optimization, mainstream diffusion models mostly remain incompetent and symbolic method adapting to dynamic goals or interactive feedback is currently scarce.

MoE for Graph and Molecule Modeling

MoE has been most successful in language models, where conditional routing significantly improves computational efficiency and model capacity. Recent studies, however, show it is also effective for graph learning, where heterogeneity in topology and scale naturally benefits from expert specialization. GraphDIVE (Hu et al. 2021) and CAME (Zhou et al. 2022) use MoE to mitigate long-tailed label distributions, while DA-MoE (Yao et al. 2024) route nodes to experts specializing in different structural patterns or receptive-field ranges. These results suggest that MoE is a suitable mechanism for handling structural diversity in graph tasks.

Reinforcement learning guided optimization

RL has long been applied to molecule generation when explicit supervision is absent and exploration is needed to discover desired properties. Early sequence-based methods such as ReLeaSE (Popova, Isayev, and Tropsha 2018) use joint training of a policy network and property predictor with the reinforce algorithm (Williams 1992) to bias SMILES generation toward targets (e.g., melting point, bioactivity), while REINVENT (Olivecrona et al. 2017) adds reward shaping to preserve chemical validity and distributional consistency.

The integration of graph neural networks with RL advanced graph-based generation: GCPN (You et al. 2018a) and MolDQN (Zhou et al. 2019) build molecule graphs in a goal-directed, multi-objective manner emphasizing validity. Most recently, the integration of RL with graph diffusion models has emerged as a promising frontier. Methods including DDPO (Black et al. 2023), DPOK (Fan et al. 2023), AlignProp (Prabhudesai et al. 2023), and DRaFT (Clark et al. 2023) model the denoising process as a multi-step Markov decision process, enabling direct policy optimization via reward signals by human feedback or downstream outcome. ELEGANT (Uehara et al. 2024a) and SEIKO (Uehara et al. 2024b) introduce entropy-regularized and low-feedback RL techniques to improve distributional fidelity and sample efficiency. GDPO (Liu et al. 2024) further improves the reinforce-based algorithm to enhance molecule generation performance indicated by target property.

Although RL-based molecule generation methods have been flourishing, most of these approaches rely on static rewards and fixed training objectives, leading to unstable model training and high output variance. In comparison, the method proposed in this work mitigates these issues by leveraging the MoE architecture to dynamically route optimization across expert policies, thereby enhancing training stability and convergence efficiency in resolving the RED task.

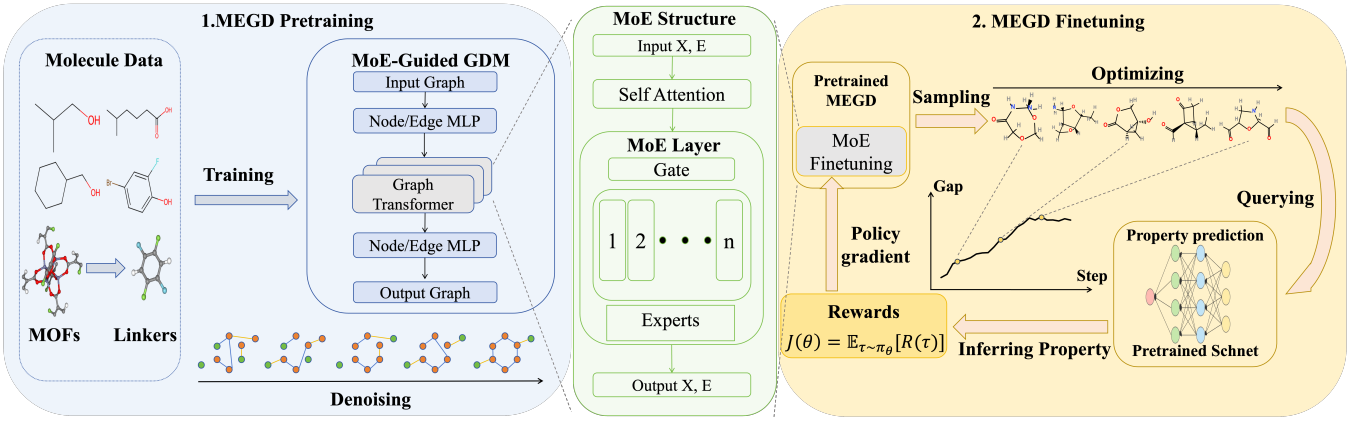

Figure 1: The MEGD pipeline is composed of two stages: the first stage (left) utilizes real-life molecule structure datasets to pre-train the GDM, ensuring its general structure generation capability; the second stage (right) applies RL to fine-tune the pre-trained model to obtain the desired structure property through iterative optimization. Our main innovation lies in the new MoE module (middle), which guides the standard graph transformer layer of GDM and receives policy gradient update from the RL stage.

Preliminaries

The task of oriented molecule design can be broadly divided into two key components: structural generation and property optimization. In this section, we provide the necessary preliminaries for both aspects.

Graph-based chemical structure design

The inherent graph structure of molecules captures the connectivity and spatial relationships between atoms, which are crucial for determining their chemical and physical properties (Katritzky et al. 2010). Molecules can be naturally represented as graphs. Denote the graph as $G = (X, E)$ with node set X and edge set E. Since the type of molecules is discrete, the feature vectors of nodes and edges are defined as one-hot vectors as $X \in R^{n\times a}$ and $E \in R^{n\times n\times b}$, which represent the atoms and chemical bonds, where n is the number of nodes, a and b are the number of types of nodes and edges (Vignac et al. 2022). The forward diffusion process gradually perturbs the original graph by randomly replacing node and edge types with other valid types, according to predefined transition probabilities. At each timestep t, the transition probabilities for nodes and edges are denoted by matrices Q_X^t and Q_E^t. These matrices govern the discrete corruption process over time, transforming the original molecule graph G into a progressively noisier version G^t.

The state transition probability can be expressed as $[Q_X^t]_{ij} = q(x^t = j|x^{t-1} = i)$ and $[Q_E^t]_{ij} = q(e^t = j|e^{t-1} = i)$. Therefore, for time step t, the joint probability of diffusion from the original graph G to the graph G^t can be defined by:

$$q(G^t|G) = \prod_{i=1}^{n} q(x_i^t|x_i, \bar{Q}_X) \prod_{1\le j<k\le n} q(e_{jk}^t|e_{jk}, \bar{Q}_E) \quad (1)$$

The goal of the model is to train a denoising network ϕ_θ parametrized by θ. The inference process uses $p_\theta(G^{t-1}|G^t)$ to sample a discrete G^{t-1} given the input graph G^t. And the objective of the model is to predict the node and the edge as accurately as possible. The training loss can be divided into $\hat{p^G} = (\hat{p^X}, \hat{p^E})$ and can be defined as:

$$\mathcal{L}(\hat{p}^G, G) = \sum_{1\le i\le n} \mathbf{CE}(x_i, \hat{p}_i^X) + \lambda \sum_{1\le i,j\le n} \mathbf{CE}(e_{ij}, \hat{p}_{ij}^E), \quad (2)$$

where CE is cross-entropy loss. When the denoising network training is completed, the original graph can be iteratively generated from the noise graph. And the type distribution of nodes and edges in the generated molecule graph will be close to the distribution in the original training dataset.

Structural properties optimization via RL

To optimize the desired molecule property, the pretrained GDM is treated as a probabilistic policy $\pi_\theta(G_0)$, parameterized by θ, which generates complete molecule graphs G_0. The training objective is defined as (Williams 1992):

$$J(\theta) = E_{G_0\sim\pi_\theta}[R(G_0)] \quad (3)$$

Here, $R(G_0)$ is a scalar value representing the performance of molecule G_0 on the target properties. RL can solve this problem very well, thus $J(\theta)$ can be optimized by policy gradient. As for the RL algorithm (Grondman et al. 2012), $\nabla_\theta J(\theta)$ is given by:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} R(G_0^{(i)}) \nabla_\theta \log \pi_\theta(G_0^{(i)}) \quad (4)$$

where N is the number of sampled molecules, and $G_0^{(i)}$ is the i-th molecule sampled from the current policy π_θ. The term $\nabla_\theta \log \pi_\theta(G_0^{(i)})$ represents the gradient of the log-probability of generating molecule $G_0^{(i)}$ under the current policy, which in a diffusion model involves the sum of log-probabilities of reverse diffusion steps:

$$\log \pi_\theta(G_0^{(i)}) = \sum_{t=1}^{T} \log p_\theta(G_{t-1}^{(i)}|G_t^{(i)}) \quad (5)$$

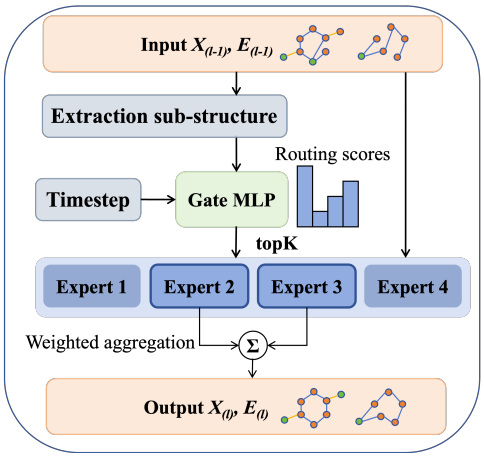

Figure 2: The MoE architecture embedded in GDMs.

By treating GDM as a probabilistic strategy and molecule property evaluation as a reward, RL algorithm enables the model to learn complex objectives such as molecule physicochemical properties and efficiently explore chemical space in a performance-driven manner. The method of dividing the diffusion process trajectory into different equivalence classes for optimization according to the difference of G_0 is called the eager policy gradient method (Liu et al. 2024), and the objective is defined as:

$$\nabla_\theta \mathcal{J}_{\mathrm{RL}} \approx \frac{1}{K}\sum_{k=1}^{K}\frac{T}{|\mathcal{T}_k|}\sum_{t\in\mathcal{T}_k} r(G_0^{(k)})\nabla_\theta \log p_\theta(G_0^{(k)}|G_t^{(k)}), \quad (6)$$

where $\mathcal{T}_k$ are random subsets of timesteps that can accelerate the estimation. Finally, the process of using this policy gradient to finetune the parameters of pretrained GDM with the learning rate η can be defined as:

$$\theta \leftarrow \theta + \eta \cdot \nabla_\theta \mathcal{J}_{\mathrm{RL}} \quad (7)$$

MoE-Guided Graph Diffusion

We introduce our method MEGD as for Fig.2, which integrates the MoE architecture into the pretraining and finetuning of GDM.

MoE in Graph Diffusion Model Pretraining

The MoE-Guided design enables the model to dynamically route input molecule graphs to a subset of specialized experts based on graph-specific features. This selective activation increases representational capacity and learning efficiency for diverse molecule structures, particularly those relevant to material design, without a linear increase in computation (Calanzone, D'Oro, and Bacon 2025).

The core idea of incorporating MoE into the GDM pretraining is to replace or augment standard layers within the denoising network with MoE layers. Each MoE layer consists of multiple independent expert networks and a gating network.

In our MoE-Guided GDM, the denoising function ϵ_θ is parameterized by a series of layers, some of which are MoE layers. For a MoE layer l, given an input feature representation $h^{(l-1)}$ which contains the node and edge features, the output $h^{(l)}$ is computed as a weighted sum of the outputs from top K expert networks. First, the gating network can be defined as:

$$g^{(l)} = Softmax\left(W_g \mathbf{h}_{t,s}^{(l-1)} + b_g\right) \in R^K, \quad (8)$$

where the output $g^{(l)}$ can represent the weight of each expert and the W_g, b_g are trainable weights and biases of the gating network. It is worth mentioning that the input of gating network is $\mathbf{h}_{t,s}$, which contains diffusion timestep t and the sub-structure vector s.

Then, each expert processes the part with the input $G^{(l-1)}$ assigned to it and gets the output $E_{l,k}(G^{(l-1)})$. The output is then weighted and summed by the weights calculated by the gating network to obtain $G^{(l)}$, which can be defined as follows:

$$G^{(l)} = \sum_{k=1}^{K} g_k^{(l)} \cdot E_{l,k}(G^{(l-1)}) \quad (9)$$

A significant problem with the MoE architecture is how to balance the load between experts (Chen et al. 2022). In order to ensure that all experts can make some contributions to the structure instead of concentrating on a certain expert, we introduced MoE-balance-loss into the pretraining of GDM to balance the load between experts, which is defined as:

$$\overline{\mathbf{p}}_{\mathrm{load}} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{p}_m, \quad (10)$$

$$\mathcal{L}_{\mathrm{balance}} = \left(\mathrm{StdDev}\left(\overline{\mathbf{p}}_{\mathrm{load}} \cdot K\right)\right)^2, \quad (11)$$

where M represents the number of MoE layers, and the purpose of reducing the balance loss is to ensure that each expert can be activated during forward propagation. According to Eq. 2, the MEGD Loss is:

$$\mathcal{L}_{\mathrm{MEGD}} = \mathcal{L}(\hat{p}^G, G) + \lambda_{balance} \cdot \mathcal{L}_{\mathrm{balance}} \quad (12)$$

We used functions from the RDKit to calculate general molecule properties, such as the quantitative evaluation of drug-likeness (QED) and synthetic accessibility (SA). QED quantifies the drug-likeness of a molecule on the basis of its physicochemical properties, while SA assesses its feasibility for chemical synthesis. These two metrics generally represent relatively common chemical properties and are quick to calculate. We calculated these metrics for the molecules to ensure the quality of the results generated during the pretraining process.

MoE in Reinforcement Learning Finetuning

Following the MoE-Guided pretraining of GDM, we proceed to finetune it by using RL. The pretrained MEGD serves as our policy network. We formally analyze the gradient updates received by both the gating network and the experts. Let $\theta = \{\theta_1, \ldots, \theta_n\}$ be the parameters of n experts and ϕ be the parameters of the gating network g^ϕ. The overall RL objective $\mathcal{J}_{\mathrm{RL}}$ decomposes into expert-specific gradients and gating gradients as follows:

$$\nabla_\theta \mathcal{J}_{expert_i} = \sum_{i=1}^{n} E_{G_0\sim p_\phi}\left[g_i^\phi(G_t)\cdot r(G_0)\cdot \nabla_{\theta_i}\log p_{\theta_i}(G_0|G_t)\right] \quad (13)$$

$$\nabla_\phi \mathcal{J}_{gate} = \sum_{i=1}^{n} E_{G_0 \sim p_\phi} \left[\nabla_\phi g_i^\phi(G_t) \cdot r(G_0) \cdot \log p_{\theta_i}(G_0|G_t) \right] \quad (14)$$

These two gradient components reflect the core learning dynamics of the MoE-Guided policy network. The expert gradient $\nabla_{\theta_i} \mathcal{J}_{expert_i}$ shows that each expert is updated in proportion to its routing probability $g_i^\phi(G_t)$ and the associated reward signal $r(G_0)$, ensuring that only a relevant subset of experts is optimized per training instance. The gating gradient $\nabla_\phi \mathcal{J}_{gate}$, on the other hand, updates the router to assign future inputs to experts that are more likely to yield high rewards. This separation of responsibilities enables targeted learning and reduces parameter interference.

As a result, the MoE-Guided architecture effectively partitions the input space and facilitates localized learning. Each expert network θ_i specializes in a narrower subregion of the state-action domain, learning smoother and less complex mappings. This reduces the variance of individual gradient estimates and improves training stability. At the same time, the gating network dynamically adapts to reward feedback, assigning more informative samples to the most relevant experts and promoting efficient policy refinement.

As for Fig.3, we show the convergence curves of different indicators of GDM and MEGD in the RL process. We can observe that thanks to the MoE architecture, MEGD can converge faster and achieve better results.

Implementation Details

In this section, we introduce some specific implementation details of our method that were not mentioned previously.

For the gating network in the MoE, we don't use the entire graph as input because doing so would be computationally expensive (Wu et al. 2020). Before training, we extract local sub-structures from the dataset and classify them into several categories. In the MoE layer, we first perform sub-structures extraction on the input, converting it into a feature vector representing each type of sub-structure before inputting it into the gating network. This effectively characterizes structural features without significantly increasing computational overhead.

For performance evaluation, we focus on chemically meaningful and structure-sensitive properties such as HOMO, gap, and dipole moment—as they are critical to specific applications. These properties require precise structural information and are evaluated using a sophisticated neural performance predictor based on the SchNet framework (Schütt et al. 2017). SMILES strings, generated by our RL-based molecule generator, are first converted into 3D molecule structures using the ETKDGv2 algorithm (Riniker and Landrum 2015). Each molecule undergoes hydrogen addition followed by geometry optimization with the MMFF94 force field (Halgren 1999) to refine its 3D atomic coordinates. Molecules that fail due to invalid SMILES, non-convergent optimization, or other structural inconsistencies (approximately 5% of inputs) are filtered out to ensure robustness. The optimized 3D geometries are then converted into ASE Atoms objects (Larsen et al. 2017) and fed into the SchNet-based predictor. This efficiency is significantly higher than traditional density functional theory

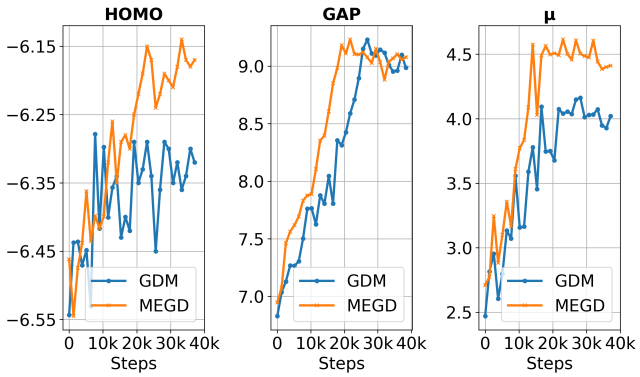

Figure 3: The comparison of three target property convergence processes between the original GDM and MEGD.

(DFT) methods, which may require hours for similar computations. By leveraging this advanced neural network, our pipeline provides accurate and differentiable property predictions, enabling the RL model to optimize molecules for specific performance criteria while maintaining computational tractability.

Experiment

We used two common datasets, QM9 and ZINC250k, and a MOF dataset to evaluate our method. We first introduce the comparison of our method with other methods on these datasets, and then we introduce some hyperparameter tuning and ablation experiments to analyze our method.

Dataset

- **QM9** comprises over 130,000 stable small organic molecules. Each molecule is composed of up to nine heavy atoms among (C, H, O, N, F).
- **ZINC250k** is a widely used benchmark for molecule generation tasks. The dataset consists of 250,000 drug-like molecules with constraints on molecular weight, logP, and synthetic accessibility to ensure realistic and synthesizable compounds.
- **QMOF** is a publicly available dataset containing quantum chemical properties of over 20,000 MOFs. We used the MOF Building Unit Developer (Halder, Prerna, and Singh 2021) to extract BUs from the dataset (Chung et al. 2019) and mixed them with the QMOF dataset as an extended dataset for our method.

Result

We compared our method with several other molecule generation methods, including MolRL-MGPT (Hu et al. 2023), GDSS (Jo, Lee, and Hwang 2022), MOOD (Lee, Jo, and Hwang 2023), DiGress (Vignac et al. 2022), DDPO (Black et al. 2023), and GDPO (Liu et al. 2024). MolRL-MGPT is a SMILES-based method, while the other methods are based on Graph Diffusion. We conducted a comprehensive performance evaluation of these methods on three datasets and three performance metrics, and reported the results for

Method	QM9			ZINC250k			QMOF		
	HOMO	Gap	μ	HOMO	Gap	μ	HOMO	Gap	μ
Full-sample									
DiffLinker	—	—	—	-9.839±2.42	7.156±2.32	8.690±4.70	-7.747±2.34	4.353±1.79	4.559±2.76
MolRL-MGPT w/o RL	-6.355±1.14	6.966±1.42	2.577±1.51	-9.471±2.77	6.097±2.50	7.368±4.36	-9.485±2.78	3.163±2.59	6.579±4.53
MolRL-MGPT	**-6.170**±1.26	8.057±1.93	3.801±1.79	-7.595±1.49	6.414±2.42	7.723±4.73	-9.119±4.97	5.457±2.41	**7.009**±3.85
GDSS	-6.402±0.73	7.470±1.69	2.464±1.29	-7.162±3.60	7.452±3.99	7.076±3.85	-7.202±1.73	4.462±2.79	4.132±3.45
MOOD	-6.346±0.72	7.633±1.95	2.580±1.49	-6.872±3.54	7.373±4.01	7.161±4.80	-7.172±1.48	4.482±3.25	4.047±3.30
DiGress	-6.442±0.62	6.744±1.29	2.855±1.47	-8.556±2.37	4.833±4.40	7.818±4.99	-7.667±1.63	4.476±2.92	4.273±2.56
DDPO	-6.396±0.58	8.890±0.89	4.224±1.78	-7.578±2.33	6.586±2.57	8.097±4.88	-7.010±2.00	5.373±2.23	5.462±4.58
GDPO	-6.302±0.54	8.110±1.17	4.011±1.86	-7.200±1.52	7.136±1.83	8.511±3.70	-6.349±1.26	4.859±2.42	5.176±3.53
MEGD w/o RL	-6.522±0.54	7.107±1.29	2.954±1.53	-8.475±2.18	5.465±3.38	8.314±3.66	-7.658±1.39	4.566±2.58	4.228±2.62
MEGD	-6.179±0.44	**9.110**±0.67	**4.473**±1.97	**-5.962**±1.17	**7.859**±1.89	**9.139**±3.65	**-6.292**±1.17	**5.620**±2.22	5.831±3.04
Top-sample									
DiffLinker	—	—	—	-5.130±0.84	10.918 ±0.88	18.898±2.44	-4.810±0.47	8.642±1.64	12.968±1.20
MolRL-MGPT w/o RL	-5.234±1.75	8.967±0.52	4.447±0.99	-3.433±2.58	10.187±0.89	16.422±2.21	-5.405±2.36	8.128±1.47	12.787±3.46
MolRL-MGPT	**-5.046**±1.94	9.837±0.66	6.688±1.34	-6.653±0.52	10.976±0.81	18.628±1.90	-6.568±1.14	8.790±1.51	12.240±3.60
GDSS	-5.444±0.41	10.021±1.51	6.052±0.83	-2.455±2.29	10.937±1.67	19.092±4.16	-5.562±1.38	7.267±1.43	10.781±3.08
MOOD	-5.457±0.61	9.963±2.18	6.222±0.97	-2.022±2.58	**11.361**±2.69	18.019±4.03	-4.999±0.97	8.438±0.80	12.303±3.46
DiGress	-5.160±0.42	9.138±0.52	5.773±0.57	-4.641±1.76	9.776±1.52	18.164±4.22	-5.319±1.08	8.084±0.89	9.004±3.09
DDPO	-4.954±0.36	10.706±0.37	8.134±1.12	-2.654±0.99	10.808±0.85	19.078±3.77	-4.299±0.84	8.796±2.15	11.286±2.78
GDPO	-4.814±0.52	10.356±0.34	8.092±1.06	-2.256±1.92	11.149±1.09	18.909±2.48	**-4.030**±0.69	8.862±2.44	11.992±3.57
MEGD w/o RL	-5.300±0.28	9.336±0.26	6.230±1.44	-4.351±1.09	10.248±1.24	18.197±4.71	-5.202±1.12	8.055±0.75	9.955±1.82
MEGD	-5.050±0.35	**10.854**±0.25	**8.433**±1.10	**-1.967**±1.76	11.296±1.16	**19.430**±1.68	-4.041±0.14	**9.055**±1.19	**13.065**±2.42

Table 1: Performance comparison on QM9, ZINC250k, and QMOF datasets. Results are reported as mean ± standard deviation. The full-sample section reports the result when sampling all molecules from the model output, while the top-sample section only reports the top 25% molecules by target property. In the DiffLinker row, their performance on the QM9 dataset is not reported due to incompatibility.

all generated molecules and higher-performing molecules. Table 1 shows that our method achieved top-2 results in 10 cases, with average performance improvements of 6.6%, 8.6%, and 8% on the three datasets, respectively. We also compared our results with other results using t-test. The results on the QM9 and ZINC250k datasets show statistical significance ($p<0.0001$ for each property) between MEGD and other methods. These results demonstrate the superiority of our method.

To evaluate the impact of integrating the MoE architecture into the GDM, we assess the generation quality across three datasets. The results in Table 2 demonstrate that MEGD maintains comparable performance to the original GDM in terms of general generation metrics such as QED and SA. Across all datasets, the changes in QED and SA are marginal and fall within the range of standard deviation, indicating that the introduction of MoE does not compromise the base model's generation capability. Notably, on the MOF dataset—which features more complex molecule structures—the MEGD exhibits a significant improvement in validity, increasing from 63.67% to 72.32%, supporting the robustness and generalizability of the MoE-Guided generation process.

We further explored the impact of changing important parameters in our method on the experimental results. We conducted experiments on the number of experts used for selection, the total number of experts, and the balance parameter of the MoE loss. Table 3 shows that in MEGD, optimal performance is achieved when choosing the right number of experts and the load balancing. Overall, compared to selecting only a single expert, all other conditions being equal, using two experts for weighted aggregation improves model performance, thanks to the model aggregating the outputs of the two best experts. A moderate total number of experts (e.g., 6) can effectively improve model performance. A low total number of experts may not achieve the desired results, while a high number may increase routing complexity and lead to performance degradation. Similarly, a moderate $\lambda_{balance}$ value (e.g., 0.05 or 0.1) can improve performance while maintaining load balance among experts; however, excessively small or large $\lambda_{balance}$ values may lead to performance degradation or uneven load. These results demonstrate that multi-expert collaboration, under appropriate balance constraints, can more effectively unleash the expressive power of the model.

We further conducted ablation experiments to evaluate the contribution of each component to our method. We compared the improvement achieved with RL finetuning across three evaluation metrics, using MEGD as the baseline. The comparison cases included: the method excluding the 'time' component, the 'sub-structure' component, and both components. The results in Fig. 4 show that removing the time component results in a 17.8% performance drop, removing the sub-structure component results in a 26.1% performance drop, and removing both results in a 41.4% performance drop. These results highlight the complementary contributions of the time and sub-structure components, demonstrating that both timestep and sub-structure guidance are essen-

Dataset	Method	QED	SA	Validity	Uniqueness	Novelty
QM9	GDM	0.487 ± 0.07	6.661 ± 0.64	99.498 ± 0.30 %	99.961 ± 0.12 %	32.198 ± 2.58 %
	MEGD	0.487 ± 0.06	6.661 ± 0.55	99.220 ± 0.30 %	99.961 ± 0.12 %	31.654 ± 2.53 %
	Difference (MEGD - GDM)	**0.000**	**0.000**	-0.278 %	**0.000** %	-0.544 %
ZINC250k	GDM	0.749 ± 0.11	5.574 ± 0.78	76.231 ± 3.45 %	100.000 ± 0.00 %	99.953 ± 0.14 %
	MEGD	0.755 ± 0.10	5.628 ± 0.63	78.320 ± 2.03 %	100.000 ± 0.00 %	100.000 ± 0.00 %
	Difference (MEGD - GDM)	**0.006**	**0.000**	**2.089** %	**0.000** %	**0.047** %
QMOF	GDM	0.450 ± 0.24	5.365 ± 1.89	63.669 ± 2.38 %	72.758 ± 1.98 %	98.076 ± 0.65 %
	MEGD	0.457 ± 0.21	5.426 ± 1.96	72.315 ± 1.52 %	76.347 ± 2.17 %	99.033 ± 0.47 %
	Difference (MEGD - GDM)	**0.007**	**0.061**	**8.646** %	**3.589** %	**0.957** %

Table 2: Comparison of generation quality between GDM and MEGD across three datasets. Performance is measured by QED, SA, Validity, Uniqueness, and Novelty. All methods are compared on their pretraining capability without applying RL.

#Selected Experts	#All Experts	λ_{balance}			
		0.01	0.05	0.1	0.5
1	4	8.642	8.985	9.068	8.897
1	6	9.069	9.064	**9.375**	9.073
1	8	9.061	8.658	**9.407**	**9.372**
2	4	9.111	9.316	**9.409**	9.234
2	6	**9.926**	**9.473**	9.230	9.181
2	8	9.080	9.103	8.969	8.357

Table 3: Experiment result on hyperparameter tuning.

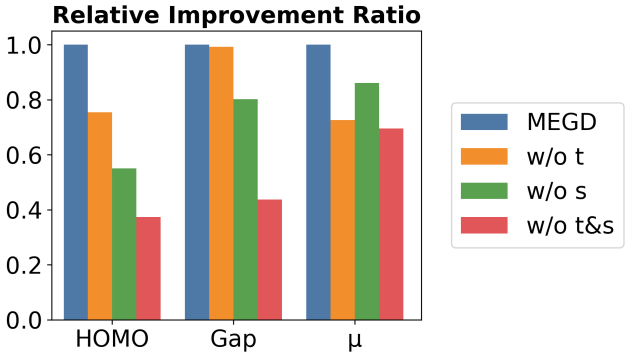

Figure 4: The ablation study result on MEGD modules.

tial for the effectiveness of MEGD.

Case Study

As shown in Fig. 5, we visualize the molecules generated by our method using Py3Dmol (Rego and Koes 2015). The results demonstrate that our MEGD model can flexibly generate molecules with diverse band gaps and effectively target those with high gaps by dynamically adjusting expert weights based on target properties under a reinforcement learning framework. This approach is particularly valuable for the design of MOFs, which consist of nodes, topologies, and linkers. Among these components, linkers provide a vast and diverse chemical space, making them especially suitable for generative modeling (Pang et al. 2025). Previous studies have shown that tuning linker properties can significantly impact MOF performance and stability (Raptopoulou

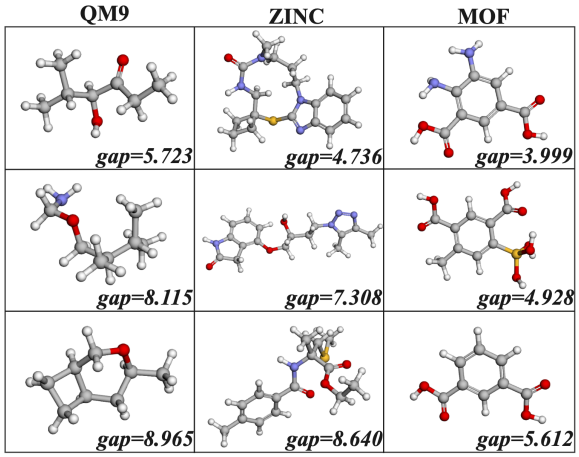

Figure 5: Visualization of molecule structures generated by MEGD on the MOFs dataset.

2021). A design paradigm that first generates linkers and then assembles them with predefined nodes and topologies has proven highly effective (Park et al. 2024). Following this paradigm, our MEGD model efficiently produces high-performance linkers and assembles them into MOFs, significantly improving molecule design efficiency and opening new avenues for the development of MOFs with enhanced stability and specialized functionalities.

Conclusion

State-of-the-art molecule structure generation paradigm depends on the cutting-edge deep learning techniques of GDM and RL, which inherently suffer from poor compatibility due to the large variance by GDM-based structure generation and the follow-up slow convergence by RL-based iterative property optimization. We propose MEGD, a new integrated molecule structure generation model that embeds the MoE architecture seamlessly into the graph diffusion process and the closed-loop of reinforcement learning for solving the oriented molecule design task. Real-world design cases on the popular material class of MOF polymers also show excellent performance of MEGD in inventing chemical and material molecule with desired properties.

Acknowledgments

This work was supported by National Key R&D Program of China (2021YFB3500700), NSFC Grant 62572026, National Social Science Fund of China 22&ZD153, State Key Laboratory of Complex & Critical Software Environment (SKLCCSE), and the Beijing Science and Technology Plan Project. Huobin Tan and Lei Shi are the corresponding authors.

References

Black, K.; Janner, M.; Du, Y.; Kostrikov, I.; and Levine, S. 2023. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*.

Calanzone, D.; D'Oro, P.; and Bacon, P.-L. 2025. Mol-MoE: Training Preference-Guided Routers for Molecule Generation. *arXiv preprint arXiv:2502.05633*.

Cao, H.; Tan, C.; Gao, Z.; Xu, Y.; Chen, G.; Heng, P.-A.; and Li, S. Z. 2024. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, 36(7): 2814–2830.

Chen, Z.; Deng, Y.; Wu, Y.; Gu, Q.; and Li, Y. 2022. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 35: 23049–23062.

Chung, Y. G.; Haldoupis, E.; Bucior, B. J.; Haranczyk, M.; Lee, S.; Zhang, H.; Vogiatzis, K. D.; Milisavljevic, M.; Ling, S.; Camp, J. S.; et al. 2019. Advances, updates, and analytics for the computation-ready, experimental metal–organic framework database: CoRE MOF 2019. *Journal of Chemical & Engineering Data*, 64(12): 5985–5998.

Clark, K.; Vicol, P.; Swersky, K.; and Fleet, D. J. 2023. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*.

Dai, D.; Deng, C.; Zhao, C.; Xu, R.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

De Cao, N.; and Kipf, T. 2018. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*.

Du, Y.; Jamasb, A. R.; Guo, J.; Fu, T.; Harris, C.; Wang, Y.; Duan, C.; Liò, P.; Schwaller, P.; and Blundell, T. L. 2024. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, 6(6): 589–604.

Fan, Y.; Watkins, O.; Du, Y.; Liu, H.; Ryu, M.; Boutilier, C.; Abbeel, P.; Ghavamzadeh, M.; Lee, K.; and Lee, K. 2023. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36: 79858–79885.

Grondman, I.; Busoniu, L.; Lopes, G. A.; and Babuska, R. 2012. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, part C (applications and reviews)*, 42(6): 1291–1307.

Halder, P.; Prerna; and Singh, J. K. 2021. Building unit extractor for metal–organic frameworks. *Journal of Chemical Information and Modeling*, 61(12): 5827–5840.

Halgren, T. A. 1999. MMFF VI. MMFF94s option for energy minimization studies. *Journal of computational chemistry*, 20(7): 720–729.

Hu, F.; Wang, L.; Wu, S.; Wang, L.; and Tan, T. 2021. Graph classification by mixture of diverse experts. *arXiv preprint arXiv:2103.15622*.

Hu, X.; Liu, G.; Zhao, Y.; and Zhang, H. 2023. De novo drug design using reinforcement learning with multiple gpt agents. *Advances in Neural Information Processing Systems*, 36: 7405–7418.

Huang, H.; Sun, L.; Du, B.; Fu, Y.; and Lv, W. 2022. Graphgdp: Generative diffusion processes for permutation invariant graph generation. In *2022 IEEE International Conference on Data Mining (ICDM)*, 201–210. IEEE.

Huang, L.; Xu, T.; Yu, Y.; Zhao, P.; Chen, X.; Han, J.; Xie, Z.; Li, H.; Zhong, W.; Wong, K.-C.; et al. 2024. A dual diffusion model enables 3D molecule generation and lead optimization based on target pockets. *Nature Communications*, 15(1): 2657.

Jin, W.; Barzilay, R.; and Jaakkola, T. 2018. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, 2323–2332. PMLR.

Jing, B.; Stärk, H.; Jaakkola, T.; and Berger, B. 2024. Generative modeling of molecular dynamics trajectories. *Advances in Neural Information Processing Systems*, 37: 40534–40564.

Jo, J.; Lee, S.; and Hwang, S. J. 2022. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International conference on machine learning*, 10362–10383. PMLR.

Katritzky, A. R.; Kuanar, M.; Slavov, S.; Hall, C. D.; Karelson, M.; Kahn, I.; and Dobchev, D. A. 2010. Quantitative correlation of physical and chemical properties with chemical structure: utility for prediction. *Chemical reviews*, 110(10): 5714–5789.

Khan, M.; Akmal, Z.; Tayyab, M.; Mansoor, S.; Zeb, A.; Ye, Z.; Zhang, J.; Wu, S.; and Wang, L. 2024. MOFs materials as photocatalysts for CO2 reduction: Progress, challenges and perspectives. *Carbon Capture Science & Technology*, 11: 100191.

Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; et al. 2017. The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27): 273002.

Lee, S.; Jo, J.; and Hwang, S. J. 2023. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, 18872–18892. PMLR.

Liu, Q.; Allamanis, M.; Brockschmidt, M.; and Gaunt, A. 2018. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31.

Liu, Y.; Du, C.; Pang, T.; Li, C.; Lin, M.; and Chen, W. 2024. Graph diffusion policy optimization. *Advances in Neural Information Processing Systems*, 37: 9585–9611.

Luo, Y.; Yan, K.; and Ji, S. 2021. Graphdf: A discrete flow model for molecular graph generation. In *International conference on machine learning*, 7192–7203. PMLR.

Madhawa, K.; Ishiguro, K.; Nakago, K.; and Abe, M. 2019. Graphnvp: An invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600*.

Maziarka, Ł.; Pocha, A.; Kaczmarczyk, J.; Rataj, K.; Danel, T.; and Warchoł, M. 2020. Mol-CycleGAN: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1): 2.

Morehead, A.; and Cheng, J. 2024. Geometry-complete diffusion for 3D molecule generation and optimization. *Communications Chemistry*, 7(1): 150.

Niu, C.; Song, Y.; Song, J.; Zhao, S.; Grover, A.; and Ermon, S. 2020. Permutation invariant graph generation via score-based generative modeling. In *International conference on artificial intelligence and statistics*, 4474–4484. PMLR.

Olivecrona, M.; Blaschke, T.; Engkvist, O.; and Chen, H. 2017. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1): 48.

Pang, J.; Jiang, W.; Zhang, X.-W.; Zhou, H.-L.; Sun, Y.; Gong, W.; Wang, B.; Ma, F.; He, L.; Chen, L.; et al. 2025. Recent progress in metal-organic frameworks (Part II—material application). *Science China Chemistry*, 68(5): 1642–1702.

Park, H.; Yan, X.; Zhu, R.; Huerta, E. A.; Chaudhuri, S.; Cooper, D.; Foster, I.; and Tajkhorshid, E. 2024. A generative artificial intelligence framework based on a molecular diffusion model for the design of metal-organic frameworks for carbon capture. *Communications Chemistry*, 7(1): 21.

Peng, X.; Guan, J.; Liu, Q.; and Ma, J. 2023. Moldiff: Addressing the atom-bond inconsistency problem in 3d molecule diffusion generation. *arXiv preprint arXiv:2305.07508*.

Popova, M.; Isayev, O.; and Tropsha, A. 2018. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7): eaap7885.

Popova, M.; Shvets, M.; Oliva, J.; and Isayev, O. 2019. MolecularRNN: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372*.

Prabhudesai, M.; Goyal, A.; Pathak, D.; and Fragkiadaki, K. 2023. Aligning text-to-image diffusion models with reward backpropagation.

Raptopoulou, C. P. 2021. Metal-organic frameworks: synthetic methods and potential applications. *Materials*, 14(2): 310.

Rego, N.; and Koes, D. 2015. 3Dmol. js: molecular visualization with WebGL. *Bioinformatics*, 31(8): 1322–1324.

Riniker, S.; and Landrum, G. A. 2015. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12): 2562–2574.

Romoff, J.; Henderson, P.; Piché, A.; Francois-Lavet, V.; and Pineau, J. 2018. Reward estimation for variance reduction in deep reinforcement learning. *arXiv preprint arXiv:1805.03359*.

Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; and Müller, K.-R. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30.

Uehara, M.; Zhao, Y.; Black, K.; Hajiramezanali, E.; Scalia, G.; Diamant, N. L.; Tseng, A. M.; Biancalani, T.; and Levine, S. 2024a. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*.

Uehara, M.; Zhao, Y.; Black, K.; Hajiramezanali, E.; Scalia, G.; Diamant, N. L.; Tseng, A. M.; Levine, S.; and Biancalani, T. 2024b. Feedback efficient online fine-tuning of diffusion models. *arXiv preprint arXiv:2402.16359*.

Vignac, C.; Krawczuk, I.; Siraudin, A.; Wang, B.; Cevher, V.; and Frossard, P. 2022. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*.

Virshup, A. M.; Contreras-García, J.; Wipf, P.; Yang, W.; and Beratan, D. N. 2013. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *Journal of the American Chemical Society*, 135(19): 7296–7303.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3): 229–256.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Xu, M.; Yu, L.; Song, Y.; Shi, C.; Ermon, S.; and Tang, J. 2022. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4): 1–39.

Yao, Z.; Liu, C.; Meng, X.; Zhan, Y.; Wu, J.; Pan, S.; and Hu, W. 2024. Da-moe: Addressing depth-sensitivity in graph-level analysis through mixture of experts. *arXiv preprint arXiv:2411.03025*.

You, J.; Liu, B.; Ying, Z.; Pande, V.; and Leskovec, J. 2018a. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31.

You, J.; Ying, R.; Ren, X.; Hamilton, W.; and Leskovec, J. 2018b. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, 5708–5717. PMLR.

Zhou, L.; Zhou, Y.; Lam, T. L.; and Xu, Y. 2022. Context-aware mixture-of-experts for unbiased scene graph generation. *arXiv preprint arXiv:2208.07109*.

Zhou, Z.; Kearnes, S.; Li, L.; Zare, R. N.; and Riley, P. 2019. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1): 10752.