MULTIVIEW CONTRASTIVE FOR GRAPH DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph diffusion models have recently emerged as powerful generative models for structured data. They learn data distributions by gradually transforming noise into samples through a reverse diffusion process, where the training objective can be interpreted as minimizing the discrepancy between generated and target distributions. From an information-theoretic perspective, we revisit the distribution-matching objective and reveal that the entropy of the target distribution is an intrinsic component of this objective. This observation indicates that preserving the information content carried by the data distribution can provide a complementary perspective for improving generative modeling. Furthermore, we show that maximizing mutual information between multiple views provides a tractable surrogate for entropy maximization. Based on this insight, we propose multiview contrastive graph diffusion learning (MCGD), a framework that introduces entropy-oriented contrastive regularization into graph diffusion models. MCGD constructs multiple views from denoised graph predictions and performs hierarchical contrastive regularization from two complementary perspectives. At the graph-level, contrastive learningis performed directly in the prediction space to preserve the structural information of generated graphs. At the semantic level, node and edge representations are jointly encoded to construct high-level graph semantics, which are then aligned across views to capture higher-order structural dependencies. By jointly constraining the prediction and semantic spaces, MCGD encourages the diffusion model to preserve both explicit graph structures and their underlying high-order semantics. Extensive experiments on general and molecular graph generation benchmarks demonstrate that MCGD improves generation quality and structural fidelity consistently.

1 INTRODUCTION

Diffusion models have recently achieved remarkable success in generative modeling and have been extended to graph generation (Ho et al., 2020; Song et al., 2021a; Vignac et al., 2023; Qin et al., 2025). Diffusion models learn generative distributions by progressively recovering clean samples from noisy states through a reverse denoising process. The denoising objective can be interpreted as minimizing the discrepancy between the generated distribution and the underlying data distribution.

From this distribution matching perspective, we further revisit the objective from an information-theoretic viewpoint. The corresponding KL divergence can be decomposed into likelihood fitting and entropy-related components. The entropy term reflects the information content carried by the data distribution. This term suggests that preserving the information content of generated distributions may provide a complementary training signal. Motivated by this observation, we introduce an entropy-inspired regularization based on multi-view contrastive learning. Specifically, we maximize the mutual information between stochastic views of denoised output, which serves as a tractable surrogate for encouraging information preservation during the generation process.

Since graph generation involves both explicit structural properties and implicit semantic characteristics, preserving information at a single level may be insufficient to fully capture the underlying graph distribution. Then, we propose a multiview contrastive graph diffusion learning (MCGD) framework that introduces multi-level contrastive constraints into graph diffusion models, as shown in Figure 1. MCGD constructs multiple views from denoised graph predictions and performs contrastive learning at two complementary levels. Graph-level contrastive learning enforces consistency in the prediction space, encouraging generated node attributes and topology to remain invariant under stochastic perturbations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Semantic-level contrastive learning further projects graph predictions into a joint latent space by integrating node and structural feature, capturing high-order graph semantics .

By jointly optimizing these objectives, MCGD complements the local denoising with information-guided structural modeling. The MCGD can be seamlessly integrated into existing graph diffusion architectures with negligible overhead, while the semantic encoder is only used during training and removed during sampling.

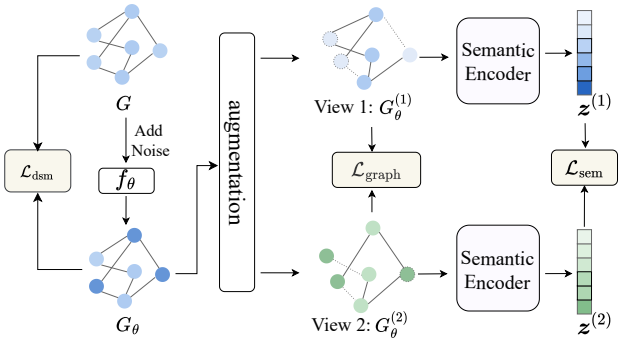


Figure 1: MCGD overview. Given an original graph G, the diffusion model f_θ generates a denoised graph G_θ. Two stochastic views $G_\theta^{(1)}$ and $G_\theta^{(2)}$ are constructed via graph augmentation. Graph-level contrastive learning $\mathcal{L}_{\text{graph}}$ enforces structural consistency, while a semantic encoder produces $z^{(1)}$ and $z^{(2)}$ for semantic-level contrastive learning $\mathcal{L}_{\text{sem}}$. The denoising objective $\mathcal{L}_{\text{dsm}}$ guides reconstruction of the original graph.

Our main contributions are summarized as follows:

(1) We provide an information-theoretic analysis of graph diffusion training and reveal that denoising objectives lack explicit constraints on information preservation. We further show that mutual information maximization provides a principled surrogate for entropy-inspired regularization.

(2) We propose MCGD, a hierarchical contrastive regularization framework that introduces graph-level and semantic-level regularization on diffusion predictions, enabling improved structural and semantic modeling.

(3) MCGD provides an entropy-inspired contrastive learning framework for graph diffusion models, promoting information preservation and structure-consistent graph generation.

Extensive experiments on general and molecular graph generation benchmarks demonstrate that MCGD improves generation fidelity and structural quality over baselines.

2 Related Work

Diffusion models have emerged as a powerful class of generative frameworks (Sohl-Dickstein et al., 2015; Ho et al., 2020), achieving remarkable success in modeling complex data distributions. Recently, diffusion-based approaches have been extended to graph generation by developing graph-aware noise processes and denoising architectures that preserve intrinsic properties such as sparsity and permutation equivariance. Existing graph generative methods can be broadly categorized into autoregressive and diffusion-based paradigms. Autoregressive approaches, such as GraphARM Kong et al. (2023), generate graph components sequentially to capture conditional dependencies, while diffusion-based methods, including GDSS Jo et al. (2022a) and DiGress Vignac et al. (2023), learn graph distributions through iterative perturbation and denoising processes. Recent studies further improve graph diffusion models through refined generation trajectories, including Beta diffusion Zhou et al. (2023), discrete flow matching Qin et al. (2025), and bridge-based diffusion processes such as GruM Jo et al. (2024), which constrain the transition paths between predefined endpoints.

Despite these advances, existing graph diffusion models mainly rely on denoising objectives derived from distribution matching to learn the underlying graph distributions. While such objectives effectively align generated and target distributions, they do not explicitly consider the information content associated with the data distribution. From an information-theoretic perspective, incorporating additional constraints on information preservation provides a complementary direction for enhancing graph generation.

A natural way to preserve informative characteristics is to maximize the mutual information (MI) between different views of the same data. Graph contrastive learning (GCL) provides an effective framework for optimizing MI-based objectives by encouraging consistency across augmented graph views. Early approaches, such as GraphCL You et al. (2020) and GRACE Zhu et al. (2020), establish

the principle of maximizing MI between graph-level and node-level augmented representations, respectively. Subsequent studies further improve contrastive learning by developing topology-aware augmentation strategies, such as GCA Zhu et al. (2021), and incorporating information-theoretic principles, such as the Information Bottleneck introduced in InfoGCL Xu et al. (2021), to preserve task-relevant information while reducing augmentation-induced noise.

From an information-theoretic perspective, maximizing MI between stochastic views has been shown to provide a tractable surrogate for entropy preservation Ma et al. (2023); Han et al. (2024). However, existing GCL methods mainly focus on representation learning over static graphs, where contrastive objectives are optimized independently from the generation process. Thus, how to integrate entropy-oriented contrastive regularization into graph diffusion models for improving information preservation remains largely unexplored.

Unlike existing contrastive learning methods that mainly use MI maximization for auxiliary representation learning, MCGD incorporates it into graph diffusion training as a hierarchical contrastive learning mechanism. Specifically, it constructs stochastic views from denoised predictions and applies Graph-Level and Semantic-level contrastive learning to preserve structural consistency and high-order semantics, respectively. This hierarchical design improves information preservation, structural consistency, and distributional fidelity during generation.

3 METHODOLOGY

3.1 CONTRASTIVE LEARNING FOR FITTING DATA DISTRIBUTION

Diffusion models aim to learn the underlying data distribution by gradually transforming random noise into samples through a predefined diffusion process. Given a clean sample $\boldsymbol{x}_0$, the VPSDE forward diffusion process progressively injects Gaussian noise: $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. This perturbation process can be equivalently described by a fixed transition kernel $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$, which determines how a clean sample is mapped into its noisy state at timestep t.

The reverse diffusion process learns a parameterized generative distribution $p_\theta(\boldsymbol{x})$ to approximate the underlying data distribution $q(\boldsymbol{x})$. Accordingly, the training objective of diffusion models can be viewed as minimizing the discrepancy between the generated and data distributions:

$$\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}\left(q(\boldsymbol{x}) \| p_\theta(\boldsymbol{x})\right). \tag{1}$$

Under the continuous-time diffusion formulation, this objective is equivalent to minimizing the Fisher divergence accumulated along the diffusion trajectory:

$$D_{\mathrm{KL}}\left(q(\boldsymbol{x})|p_\theta(\boldsymbol{x})\right) = \frac{1}{2}\int_0^T g_t^2 D_F\left(q(\boldsymbol{x}_t)|p_\theta(\boldsymbol{x}_t)\right)\mathrm{d}t, \tag{2}$$

where $q(\boldsymbol{x}_t)$ and $p_\theta(\boldsymbol{x}_t)$ denote the perturbed marginal distributions at diffusion timestep t.

Minimizing the Fisher divergence aligns the generated and data noisy marginals $p_\theta(\boldsymbol{x}_t)$ and $q(\boldsymbol{x}_t)$ throughout the diffusion trajectory. Because the reverse diffusion model reconstructs clean samples by repeatedly applying the learned reverse transition from noisy states to clean states, such marginal alignment implicitly constrains the denoising trajectory. Thus, improving the consistency of noisy-state distributions ultimately promotes the recovery of a clean sample distribution closer to the target data distribution, i.e., $p_\theta(\widehat{\boldsymbol{x}}_0) \approx q(\boldsymbol{x})$.

While the this objective enforces distribution matching through likelihood optimization, we revisit it from an information-theoretic perspective. Expanding the KL divergence gives

$$D_{\mathrm{KL}}\left(q\|p_\theta\right) = -\mathcal{H}(\boldsymbol{x}) - \mathbb{E}_{q(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x})\right] \geqslant 0, \tag{3}$$

where $\mathcal{H}(\boldsymbol{x})$ denotes the entropy of the data distribution.

Eq. 3 reveals that the entropy of the data distribution is an intrinsic component of the distribution-matching objective. Although this entropy term is independent of the model parameters, it suggests that preserving the information content of the data distribution may provide a complementary training signal beyond likelihood matching. Motivated by this observation, we introduce an entropy-oriented regularization through multiview contrastive learning.

In multiview contrastive learning, given the denoised output $\boldsymbol{x}_\theta$ generated by the diffusion model, two stochastic views are constructed through independent augmentation operators, denoted as $\boldsymbol{x}_\theta^{(1)}$ and $\boldsymbol{x}_\theta^{(2)}$, generally satisfy conditional independence given the denoised output $\boldsymbol{x}$. This forms a Markov chain $\boldsymbol{x}_\theta^{(1)} \leftrightarrow \boldsymbol{x} \leftrightarrow \boldsymbol{x}_\theta^{(2)}$, satisfying $\boldsymbol{x}_\theta^{(1)} \perp \boldsymbol{x}_\theta^{(2)} \mid \boldsymbol{x}$. Followingthe properties of information theory, the information entropy of discrete data is numerically equivalent to its self-mutual information, $\mathcal{H}(\boldsymbol{x}) = \mathcal{I}(\boldsymbol{x}; \boldsymbol{x})$. Then, we have the inequality (Ma et al., 2023; Han et al., 2024):

$$\mathcal{H}(\boldsymbol{x}) = \mathcal{I}(\boldsymbol{x}; \boldsymbol{x}) \geqslant \mathcal{I}(\boldsymbol{x}_\theta^{(1)}; \boldsymbol{x}_\theta^{(2)}) \,. \tag{4}$$

Thus, maximizing the mutual information between stochastic views provides a tractable surrogate for entropy maximization, encouraging the generated distribution to preserve informative structural characteristics beyond likelihood matching.

3.2 Graph-Level Contrastive Learning

The mutual information between these two augmented views $\mathcal{I}(\boldsymbol{x}^{(1)}; \boldsymbol{x}^{(2)})$ can be defined via the Kullback-Leibler (KL) divergence in information theory:

$$\mathcal{I}(\boldsymbol{x}^{(1)}; \boldsymbol{x}^{(2)}) = D_{\mathrm{KL}}\big(p(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \| p(\boldsymbol{x}^{(1)}) p(\boldsymbol{x}^{(2)})\big) \,. \tag{5}$$

To overcome the training bias and optimization instability caused by the asymmetry of the KL divergence, the Jensen-Shannon (JS) divergence is typically employed as a symmetric alternative. Incorporating contrastive learning theory, by introducing a network $s_\theta(\cdot, \cdot)$ parameterized by θ, we can derive the variational lower bound of the JS divergence:

$$\begin{aligned} &\mathrm{D}_{\mathrm{JS}}\big(p(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \| p(\boldsymbol{x}^{(1)}) p(\boldsymbol{x}^{(2)})\big) \\ &\quad \geqslant \mathbb{E}_{p(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})} \log\big(s_\theta(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})\big) + \mathbb{E}_{p(\boldsymbol{x}^{(1)}) p(\boldsymbol{x}^{(2)})} \log\big(1 - s_\theta(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})\big) \,. \end{aligned} \tag{6}$$

Following the variational divergence minimization framework (Nowozin et al., 2016; Oord et al., 2018; Ma et al., 2023; Han et al., 2024), we transform this intractable JS divergence objective Eq. 6 into a lower bound estimate based on a discriminator $s(\cdot, \cdot)$:

$$\mathcal{I}(\boldsymbol{x}_\theta^{(1)}; \boldsymbol{x}_\theta^{(2)}) \geqslant \mathbb{E}_{\mathcal{X}_{po}}[\log s(\boldsymbol{x}_\theta^{(1)}; \boldsymbol{x}_\theta^{(2)})] + \mathbb{E}_{\mathcal{X}_{ne}}[\log(1 - s(\boldsymbol{x}_\theta^{(1)}; \boldsymbol{x}_\theta^{(2)}))], \tag{7}$$

where $\mathcal{X}_{po}$ and $\mathcal{X}_{ne}$ denote the sets of positive and negative sample pairs, respectively, $\boldsymbol{x}_\theta$ denotes diffusion denoising model θ output of the feature $\boldsymbol{x}$.

To render this abstract objective computationally efficient, we parameterize the discriminator by assuming the observation bias follows a Gaussian distribution (Ma et al., 2023; Han et al., 2024). This yields a Gaussian form for the discriminator:

$$s(\boldsymbol{x}_\theta^{(1)}; \boldsymbol{x}_\theta^{(2)}) = \exp\left(-\|\boldsymbol{x}_\theta^{(1)} - \boldsymbol{x}_\theta^{(2)}\|_2^2\right). \tag{8}$$

In this diffusion process, the reconstruction loss naturally ensures that non-corresponding samples are well-separated in the prediction space. Inspired by the philosophy of Bootstrap Your Own Latent (Grill et al., 2020; Ma et al., 2023; Han et al., 2024), we observe that he Gaussian kernel exhibits exponential decay of the Gaussian kernel, the gradient contribution from negative samples becomes negligible when the distance between views is large. By treating these terms as constant biases and omitting them from the optimization objective Eq. 7, we derive the structural consistency loss in the prediction space:

$$\mathcal{L}_{\mathrm{graph}} = \frac{1}{n_b} \sum_{j=1}^{n_b} \big(\|X_{j,\theta}^{(1)} - X_{j,\theta}^{(2)}\|_{\mathrm{F}}^2 + \|A_{j,\theta}^{(1)} - A_{j,\theta}^{(2)}\|_{\mathrm{F}}^2\big). \tag{9}$$

Thus, constructing meaningful stochastic views from denoised graph predictions is crucial for preserving structural information through graph-level contrastive learning.

3.3 Semantic-Level Contrastive Learning

Beyond enforcing consistency in the prediction space, we further encourage semantic alignment between augmented graph views through a Semantic Encoder. As shown in Figure 1, we first generate

two augmented views node features $X_\theta^{(i)}$ and adjacency matrices $A_\theta^{(i)}$, where $i \in \{1, 2\}$ denote two augmented views based on the denoised graph outputs X_θ and A_θ. To capture complementary graph semantics, semantic encoder employs dedicated encoders to separately model attribute and structural information. Specifically, the node encoder f_ψ maps node attributes into a latent feature space: $\boldsymbol{h}_x^{(i)} = f_\psi(X_\theta^{(i)})$, while the edge encoder f_ϕ captures structural information from the adjacency matrix: $\boldsymbol{h}_{\text{adj}}^{(i)} = f_\phi(A_\theta^{(i)})$. We fuse attribute and structural representations through feature concatenation to form a unified graph descriptor, which is subsequently transformed by a nonlinear projection head: $\boldsymbol{z}^{(i)} = g_\xi([\boldsymbol{h}_x^{(i)}, \boldsymbol{h}_{\text{adj}}^{(i)}])$, where g_ξ denotes the projection head aligns heterogeneous representations into a unified semantic latent space, facilitating Semantic-level contrastive learning. By jointly integrating attribute semantics and topological structures, the resulting embedding captures a more comprehensive graph representation than either modality alone, providing a robust semantic basis for contrastive learning.

Algorithm 1: MCGD Training Algorithm

Input: $\mathcal{D}$, f_θ, f_ψ, f_ϕ, g_ξ, and $\text{aug}(\cdot)$.
Initialize: $\theta^\star$.
for Sample n_b graphs G from $\mathcal{D}$ **do**
  $t \sim \text{Uniform}(1, T)$ and $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$;
  Generate noisy graph G_t by forward diffusion;
  $G_\theta \leftarrow f_\theta(G_t, t)$;
  $G_\theta^{(1)}, G_\theta^{(2)} \leftarrow \text{aug}(G_\theta), \text{aug}(G_\theta)$;
  $\boldsymbol{h}_x^{(i)} \leftarrow f_\psi(X_\theta^{(i)})$, $\boldsymbol{h}_{\text{adj}}^{(i)} \leftarrow f_\phi(A_\theta^{(i)})$, $\forall i \in \{1, 2\}$;
  $\boldsymbol{z}^{(i)} \leftarrow g_\xi([\boldsymbol{h}_x^{(i)}, \boldsymbol{h}_{\text{adj}}^{(i)}])$, $\forall i \in \{1, 2\}$;
  Calculate $\mathcal{L}_{\text{dsm}}$ by Eq. 13;
  Calculate $\mathcal{L}_{\text{graph}}$ by Eq. 9;
  Calculate $\mathcal{L}_{\text{sem}}$ by Eq. 12;
  $\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{dsm}} + \lambda_g \mathcal{L}_{\text{graph}} + \lambda_s \mathcal{L}_{\text{sem}}$;
  Update $\{\theta, \psi, \phi, \xi\}$ by minimizing $\mathcal{L}_{\text{overall}}$;
end for
Output: $\theta^\star = \theta$.

Based on our previous theorem, maximizing this mutual information term effectively maximizes the lower bound of the data's information entropy. However, directly computing and optimizing exact mutual information in high-dimensional continuous spaces parameterized by neural networks is computationally intractable. To address this, contrastive learning introduces a tractable variational lower bound of mutual information (Oord et al., 2018; Ma et al., 2023; Han et al., 2024). Specifically, maximizing mutual information can be equivalently approximated by maximizing its lower bound. For a network $s_\theta(\cdot, \cdot)$ that measures the compatibility between two views, the InfoNCE loss provides a lower bound: $\mathcal{I}(\boldsymbol{x}^{(1)}; \boldsymbol{x}^{(2)}) \geq \log(|\mathcal{X}_{ne}|) - L_{\text{cl}}$, where $|\mathcal{X}_{ne}|$ is the number of negative samples. The lower bound can be transformed into a minimization problem over a contrastive objective.

A widely used formulation is the InfoNCE loss (Oord et al., 2018; Chen et al., 2020), which has been shown to provide a tight lower bound on mutual information. To learn discriminative graph representations, we introduce the InfoNCE loss (Oord et al., 2018; Chen et al., 2020). Specifically, given a minibatch of n_b samples, we set a sample is $\boldsymbol{x}_i$, we perform two independent stochastic augmentations on each sample to obtain $2n_b$ augmented views. Two views derived from the same original sample form a positive pair $(\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)})$, while the remaining $2(n_b - 1)$ augmented samples in the minibatch are treated as negative samples. The loss for i-th positive pair is defined as follows:

$$\ell(\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}) = -\log \frac{\exp(\text{sim}(\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)})/\tau)}{\sum_{j=1,2} \sum_{k \neq i}^{n_b} \exp(\text{sim}(\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_k^{(j)})/\tau)}, \tag{10}$$

where τ denotes a temperature parameter, and $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity between two feature vectors. Then, we define the following contrastive loss:

$$L_{\text{cl}}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) = \frac{1}{2}\left(\ell(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) + \ell(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)})\right). \tag{11}$$

Minimizing the InfoNCE loss is equivalent to maximizing a lower bound of the mutual information between augmented views. Thus, optimizing Eq. 11 effectively encourages the model to capture shared information across views, which in turn approximates entropy maximization under the information-theoretic framework. This perfectly aligns with our theoretical motivation to use contrastive learning as a hierarchical information entropy regularizer.

In this semantic latent space, positive pairs are constructed from two augmented views of the same graph, while representations from different graphs are treated as negatives. By maximizing the

mutual information $\mathcal{I}(\boldsymbol{z}^{(1)}; \boldsymbol{z}^{(2)})$ in this joint semantic space, the model constructs a global semantic anchor. This mechanism forces the latent space to preserve the overall topological identity, effectively bridging the gap where pointwise regression fails to capture complex global dependencies. The semantic-level contrastive learing loss is given by:

$$\mathcal{L}_{\text{sem}} = \sum_{i=1}^{n_b} L_{\text{cl}}(\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)}), \tag{12}$$

where L_{cl} denotes the InfoNCE loss (Oord et al., 2018; Chen et al., 2020) defined in Eq. 11.

3.4 MCGD Algorithm

To effectively capture the intrinsic structure of graphs while maintaining denoising precision, we propose a multiview contrastive graph diffusion learning (MCGD) method as shown in Figure 1. The training process of MCGD operates by jointly optimizing the denoising reconstruction objective Eq. 13 with hierarchical contrastive learning across both the prediction and representation spaces.

$$\mathcal{L}_{\text{dsm}} = \mathbb{E}_{t,\text{G}_0,\text{G}_\text{t}} \left[\|X - X_\theta\|^2 + \|A - A_\theta\|^2 \right]. \tag{13}$$

We sum the hierarchical contrastive losses to obtain the overall loss:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{dsm}} + \lambda_g \mathcal{L}_{\text{graph}} + \lambda_s \mathcal{L}_{\text{sem}}, \tag{14}$$

where λ_g, and λ_s are trade-off parameters for the graph- and semantic-level constraints, respectively.

As summarized in Algorithm 1, MCGD follows a structured multi-stage optimization pipeline. At each training iteration, a clean graph sample $G = (X, A)$ is first drawn from the empirical data distribution. A timestep t is uniformly sampled and Gaussian noise is injected through the forward diffusion process to obtain the corrupted graph G_t. f_θ then predicts the reconstructed graph G_θ, which serves as the foundation for subsequent hierarchical regularization objectives.

To construct contrastive supervision, two stochastic augmentations are independently applied to G_θ, generating two correlated views $G_\theta^{(1)}$ and $G_\theta^{(2)}$.

At the graph level, structural consistency is enforced in the prediction space by minimizing the discrepancy between the two augmented reconstructions. $\mathcal{L}_{\text{graph}}$ ensures that the global topology remains stable under stochastic transformations, acting as a coarse-grained structural anchor.

At the semantic level, the node and edge representations are fused through a projection head g_ξ to produce joint embeddings $\boldsymbol{z}^{(i)}$. By maximizing the mutual information between $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$, the semantic-level loss $\mathcal{L}_{\text{sem}}$ captures high-order dependencies between node attributes and graph topology, preserving semantic consistency.

All objectives are integrated into a unified loss function $\mathcal{L}_{\text{overall}}$. The model weights $\{\theta, \psi, \phi, \xi\}$ are jointly optimized by gradient-based methods. Through this hierarchical design, MCGD simultaneously preserves local fidelity and structural integrity, effectively preserving both structural consistency and semantic information, allowing diffusion models to better capture complex graph distributions.

4 Experiments

4.1 Experimental Setting

In Semantic Encoder module, we introduce a lightweight semantic encoder operating on the enhanced graph predictions. The node encoder f_ψ maps predicted node attributes into a latent representation and aggregates node information through masked global average pooling. In parallel, the edge encoder f_ϕ projects the flattened adjacency matrix into a structural embedding. The resulting node and topology representations are concatenated and transformed by a projection network g_ξ to obtain a 128-dimensional semantic representation used for contrastive learning.

By jointly encoding attribute-level and topology-level information, the semantic encoder facilitates the learning of invariant graph semantics under stochastic augmentations. The semantic encoder

Table 1: Comparison of MCGD with baselines on Planar and SBM.

Method	Planar				SBM			
	Synthetic, $\|V\| = 64$				Synthetic, $44 \leq \|V\| \leq 187$			
	Deg. ↓	Clus. ↓	Orbit ↓	Spec. ↓	Deg. ↓	Clus. ↓	Orbit ↓	Spec. ↓
Training set	0.0002	0.0310	0.0005	0.0052	0.0008	0.0332	0.0255	0.0063
GraphRNN	0.0049	0.2779	1.2543	0.0459	0.0055	0.0584	0.0785	0.0065
GRAN	0.0007	0.0426	0.0009	0.0075	0.0113	0.0553	0.0540	0.0054
SPECTRE	0.0005	0.0785	0.0012	0.0112	0.0015	0.0521	0.0412	0.0056
EDP-GNN	0.0044	0.3187	1.4986	0.0813	0.0011	0.0552	0.0520	0.0070
GDSS	0.0041	0.2676	0.1720	0.0370	0.0212	0.0646	0.0894	0.0128
ConGress	0.0048	0.2728	1.2950	0.0418	0.0273	0.1029	0.1148	–
DiGress	0.0007	0.0780	0.0079	0.0098	0.0018	0.0485	0.0415	0.0045
GruM	0.0007	0.0489	0.0048	0.0075	0.0007	0.0492	0.0448	0.0050
DeFoG	0.0005	0.0501	0.0006	0.0072	0.0006	0.0517	0.0556	0.0054
MCGD	0.0003	0.0194	0.0032	0.0030	0.0003	0.0280	0.0245	0.0031

Table 2: Comparison of MCGD with baselines on Ego-small and Enzymes.

Method	Ego-small				Enzymes			
	Real, $4 \leq \|V\| \leq 18$				Real, $10 \leq \|V\| \leq 125$			
	Deg.↓	Clus.↓	Orbit↓	Avg.↓	Deg.↓	Clus.↓	Orbit↓	Avg.↓
GraphRNN	0.090	0.220	0.003	0.104	0.017	0.062	0.046	0.042
GraphAF	0.030	0.110	0.001	0.047	1.669	1.283	0.266	1.073
GraphDF	0.040	0.130	0.010	0.060	1.503	1.061	0.202	0.922
GraphVAE	0.130	0.170	0.050	0.117	1.369	0.629	0.191	0.730
GNF	0.030	0.100	0.001	0.044	-	-	-	-
EDP-GNN	0.052	0.093	0.007	0.051	0.023	0.268	0.082	0.124
GPrinFlowNet	-	-	-	-	0.021	0.088	0.009	0.039
SPECTRE	0.078	0.078	0.007	0.054	0.136	0.195	0.125	0.152
GDSS	0.021	0.024	0.007	0.017	0.026	0.061	0.009	0.032
DiGress	0.015	0.029	0.005	0.016	0.004	0.083	0.002	0.030
HOG-Diff	0.015	0.027	0.004	0.016	0.011	0.061	0.007	0.027
MCGD	0.0066	0.0302	0.0047	0.0138	0.0065	0.0346	0.0111	0.0174

module is applied only to the predicted clean graph representations after the denoising network and does not participate in the diffusion backbone or reverse sampling process. Two semantic views are constructed by applying weak and strong stochastic augmentations to the predicted graph, where the augmentations introduce feature dropout and Gaussian perturbations on the predicted node attributes and adjacency structures. Overall, MCGD incurs negligible computational overhead. The semantic encoder is employed only during training and discarded during graph generation and sampling. For graph denoising, we employ a Graph Transformer-based denoising network f_θ, building upon the architectures of Dwivedi & Bresson (2020) and Vignac et al. (2022), with higher-order adjacency information (Jo et al., 2022b) and FiLM-based attention mechanisms (Perez et al., 2018).

The model is optimized using AdamW with exponential moving average. During sampling, we employ an Euler-Maruyama solver together with predictor-corrector refinement. Additional implementation details are provided in Appendix C. The network architectures are described in Appendix B. Data augmentation strategies are detailed in Appendix A. The sensitivity analysis of the weighting coefficients for graph-level λ_g and semantic-level λ_s constraints is presented in Appendix D. Dataset-specific details are provided in Appendix E.

Table 3: Comparison of MCGD with baselines on QM9 and ZINC250k.

Method	QM9				ZINC250k			
	Real, $\|V\| \leq 9$				Synthetic, $\|V\| \leq 38$			
	Valid (%) ↑	FCD ↓	NSPDK ↓	Scaf. ↑	Valid (%) ↑	FCD ↓	NSPDK ↓	Scaf. ↑
Training set	100.0	0.0398	0.0001	0.9719	100.0	0.0615	0.0001	0.8395
MoFlow	91.36	4.4670	0.0169	0.1447	63.11	20.931	0.0455	0.0133
GraphAF	74.43	5.6250	0.0207	0.3046	68.47	16.023	0.0442	0.0672
GraphDF	93.88	10.928	0.0636	0.0978	90.61	33.546	0.1770	0.0000
EDP-GNN	47.52	2.6800	0.0046	0.3270	82.97	16.737	0.0485	0.0000
GDSS	95.72	2.9000	0.0033	0.6983	97.01	14.656	0.0195	0.0467
DiGress	99.59	0.3000	0.0006	0.9267	94.99	3.4820	0.0021	0.4163
GraphARM	90.25	1.2200	0.0020	-	88.23	16.260	0.0550	-
GruM	99.79	0.1238	0.0003	0.9357	98.65	2.2570	0.0015	0.5299
DeFoG	99.26	0.2680	0.0005	-	94.97	2.0300	0.0020	-
HOG-Diff	98.74	0.1720	0.0003	-	98.56	1.6330	0.0010	-
MCGD	99.61	0.1054	0.0003	0.9367	99.47	1.5794	0.0012	0.5367

4.2 General Graph Generation

The experimental results are summarized in Tables 1 and 2.

Datasets and Metrics. Our experiments involve four benchmark datasets: Planar Dobson & Doig (2003); Vignac et al. (2023), Stochastic Block Model (SBM) Dobson & Doig (2003); Vignac et al. (2023), Ego-small Sen et al. (2008), Enzymes Schomburg et al. (2004). Planar consists of planar graphs satisfying non-crossing edge constraints. SBM contains graphs generated from stochastic block models with community structures. Ego-small is composed of ego networks extracted from the Citeseer citation graph. Enzymes contains 587 protein graphs representing the tertiary structures of enzymes collected from the BRENDA database. Dataset statistics are provided in Appendix E.

We evaluate graph generation quality using Maximum Mean Discrepancy (MMD) You et al. (2018) between generated and test graph statistics. For Planar and SBM, we measure degree (Deg.), clustering coefficient (Clus.), Orbit, and Spectral statistics(Spec.), while degree (Deg.), clustering coefficient (Clus.), Orbit are used for Ego-small and Enzymes. We also report the average MMD (Avg. MMD) over all statistics for distribution alignment. Lower MMD indicates better alignment.

Baselines. For Planar and SBM, we compare MCGD against a diverse set of representative graph generative models, including GraphRNN You et al. (2018), GRAN Liao et al. (2019), SPECTRE Martinkus et al. (2022), EDP-GNN Niu et al. (2020), GDSS Jo et al. (2022a), ConGress Vignac et al. (2023), DiGress Vignac et al. (2023), GruM Jo et al. (2024), and DeFoG Qin et al. (2025). For Ego-small and Enzymes, we compare MCGD with GraphRNN You et al. (2018), GraphAF Shi et al. (2020), GraphDF Luo et al. (2021), GraphVAE Simonovsky (2018), GNF Liu et al. (2019), EDP-GNN Niu et al. (2020), GPrinFlowNet Luo et al. (2023), SPECTRE Martinkus et al. (2022), GDSS Jo et al. (2022a), DiGress Vignac et al. (2023) and HOG-DiffHuang & Birdal (2026).

Results. On Planar and SBM, MCGD achieves strong clustering, orbit, and spectral statistics, while on Ego-small and Enzymes it reduces degree and overall structural discrepancies. These results demonstrate robust preservation of local connectivity and global structure across synthetic and real-world datasets. Visualizations are provided in Appendix F.

4.3 Molecular Graph Generation

The experimental results are summarized in Table 3.

Datasets and Metrics. We evaluate our model on molecular benchmarks: QM9 (Ramakrishnan et al., 2014) and ZINC250k (Irwin et al., 2012). Detailed descriptions are provided in Appendix E. To comprehensively and rigorously assess the generative quality, we employ four complementary metrics:

Table 4: Ablation study of different loss terms on QM9 and Planar.

Variant	QM9				Planar			
	Real, $\|V\| \leq 9$				*Synthetic*, $\|V\| = 64$			
	Valid (%) ↑	FCD ↓	NSPDK ↓	Scaf. ↑	Deg. ↓	Clus. ↓	Orbit ↓	Spec. ↓
Baseline	99.79	0.1238	0.0003	0.9357	0.0007	0.0489	0.0048	0.0075
w/o $\mathcal{L}_{\text{graph}}$	99.76	0.1208	0.0003	0.9424	0.0002	0.0220	0.0035	0.0050
w/o $\mathcal{L}_{\text{sem}}$	99.64	0.1069	0.0003	0.9416	0.0005	0.0322	0.0032	0.0034
MCGD	99.61	0.1054	0.0003	0.9367	0.0003	0.0194	0.0032	0.0034

(1) Validity calculates the percentage of generated molecules that adhere to chemical valence rules without post-hoc correction; (2) Fréchet ChemNet Distance (FCD) (Preuer et al., 2018) measures the distributional discrepancy between generated and reference sets in the deep chemical representation space; (3) NSPDK MMD (Costa & Grave, 2010) evaluates the topological similarity based on neighborhood subgraph kernels; and (4) Scaffold Similarity (Scaf.) quantifies the preservation of essential molecular backbones and functional substructures.

Baselines. To evaluate our MCGD, we compare it against a diverse set of representative molecular graph generation models, including MoFlow (Zang & Wang, 2020), GraphAF, GraphDF (Luo et al., 2021), EDP-GNN, GDSS, DiGress, GraphARM Kong et al. (2023), GruM (Jo et al., 2024), DeFoG (Qin et al., 2025), and HOG-Diff.

Results. MCGD achieves strong molecular generation performance on both QM9 and ZINC250k, with high validity and competitive FCD, NSPDK, and scaffold scores, indicating its ability to model chemically valid structures, molecular distributions, local subgraph patterns, and diverse scaffolds. These results demonstrate that hierarchical consistency regularization captures both structural and semantic information beyond conventional denoising-based generation. Visualizations of MCGD are provided in Appendix F.

4.4 Ablation studies

We evaluate the contributions of $\mathcal{L}_{\text{graph}}$ and $\mathcal{L}_{\text{sem}}$ on QM9 and Planar, with results in Table 4. Each component improves generation quality, while their combination achieves the best performance, demonstrating complementary benefits. Specifically, $\mathcal{L}_{\text{graph}}$ enhances structural consistency, whereas $\mathcal{L}_{\text{sem}}$ captures high-level semantic information.

5 Conclusion

We propose MCGD, an information-theoretic regularization framework for graph diffusion models. We revisit the conventional distribution-matching objective from an information-theoretic perspective and show that entropy provides an intrinsic perspective for characterizing the information content of the target distribution. Based on this observation, we establish a connection between entropy-oriented regularization and mutual information maximization, and use contrastive learning between stochastic graph views as a tractable surrogate.

MCGD further realizes this principle through two complementary levels of contrastive regularization: graph-level contrastive learning preserves structural information directly in the prediction space, while semantic-level contrastive learning captures higher-order dependencies in a joint node-edge semantic space. By complementing pointwise diffusion objectives with explicit information-oriented regularization, MCGD encourages the generated graphs to preserve both explicit structural patterns and high-level graph semantics. Extensive experiments on general and molecular graph generation benchmarks demonstrate that MCGD consistently improves generation quality and structural fidelity over existing graph diffusion models. Overall, These results highlight the potential of multi-view contrastive learning as a principled and effective way to enhance the structural awareness of diffusion-based graph generation.

AI USE STATEMENT

Generative AI tools were used for language polishing. All AI-assisted content was carefully checked and revised by the authors, who are responsible for the accuracy and integrity of the final manuscript.

ETHICS STATEMENT

Our experiments rely exclusively on publicly available datasets and established evaluation protocols. No human participants, personally identifiable information, or other sensitive personal data are involved in this study. All datasets and evaluation resources are used in accordance with their respective terms and attribution requirements. This work follows the applicable ethical guidelines of the conference.

REPRODUCIBILITY STATEMENT

We report the key implementation details, training configurations, hyperparameters, datasets, and evaluation procedures required to reproduce our experiments. To further support reproducibility, the source code and relevant configuration and evaluation scripts will be released upon acceptance or with the camera-ready submission.

REFERENCES

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119, pp. 1597–1607, 2020.

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *ICML*, volume 10, pp. 255–262, 2010.

Paul D. Dobson and Andrew J. Doig. Distinguishing Enzyme structures from non-Enzymes without alignments. *Journal of Molecular Biology*, 330 4:771–83, 2003.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, volume 33, pp. 21271–21284, 2020.

Qi Han, Zhibo Tian, Chengwei Xia, and Kun Zhan. InfoMatch: Entropy neural estimation for semi-supervised image classification. In *IJCAI*, volume 33, pp. 4089–4097, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pp. 6840–6851, 2020.

Yiming Huang and Tolga Birdal. HOG-Diff: Higher-order guided diffusion for graph generation. In *ICLR*, volume 14, 2026.

John J. Irwin, T. 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: 1757–1768, 2012.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *ICML*, volume 162, pp. 10362–10383, 2022a.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *ICML*, pp. 10362–10383, 2022b.

Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. In *ICML*, volume 235, pp. 22371–22405, 2024.

Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B. Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In *ICML*, volume 202, pp. 17391–17408, 2023.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *NeurIPS*, volume 32, 2019.

Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph normalizing flows. In *NeurIPS*, volume 32. Curran Associates, Inc., 2019.

Tianze Luo, Zhanfeng Mo, and Sinno Jialin Pan. Conditional graph generation with graph principal flow network. In *ICML*, 2023.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. GraphDF: A discrete flow model for molecular graph generation. In *ICML*, volume 139, pp. 7192–7203, 2021.

Yixuan Ma, Xiaolin Zhang, Peng Zhang, and Kun Zhan. Entropy neural estimation for graph contrastive learning. In *ACM MM*, volume 31, pp. 435–443, 2023.

Karolis Martinkus, Andreas Loukas, Nathanael Perraudin, and Roger Wattenhofer. SPECTRE : Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In *ICML*, volume 162, pp. 15159–15179, 2022.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *AISTATS*, pp. 4474–4484, 2020.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *NeurIPS*, volume 29, pp. 271–279, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, volume 32, 2018.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. DeFoG: Discrete flow matching for graph generation. In *ICML*, volume 267, pp. 50269–50326, 2025.

Raghunathan Ramakrishnan, Pavlo O. Dral, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. BRENDA, the enzyme database: updates and major new developments. *Nucleic Acids Research*, 32 Database issue:D431–3, 2004.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *The AI Magazine*, 2008.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: a flow-based autoregressive model for molecular graph generation. In *ICLR*, volume 8, 2020.

Martin Simonovsky. GraphVAE: Towards generation of small graphs using variational autoencoders. In *ICANN*, volume 11139, pp. 412–422, 2018.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, volume 37, pp. 2256–2265, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, volume 9, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021b.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *ICLR*, volume 11, pp. 1–13, 2023.

Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. InfoGCL: Information-aware graph contrastive learning. In *NeurIPS*, volume 34, pp. 30414–30425, 2021.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *ICML*, volume 80, pp. 5708–5717, 2018.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, volume 33, pp. 5812–5823, 2020.

Chengxi Zang and Fei Wang. MoFlow: An invertible flow model for generating molecular graphs. In *KDD*, volume 20, pp. 617–626, 2020.

Mingyuan Zhou, Tianqi Chen, Zhendong Wang, and Huangjie Zheng. Beta diffusion. In *NeurIPS*, volume 36, pp. 30070–30095, 2023.

Yanqiao Zhu, Yichen Cao, Weizhuo Guo, Bingzhe Qu, Jie Gu, and Liang Wang. Deep graph contrastive representation learning. In *ICML*, 2020.

Yanqiao Zhu, Feng Xu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW*, pp. 2069–2080, 2021.

Appendix

This supplementary material provides additional information to support the main text, including data Augmentation Strategies, implementation details and hyperparameters, dataset descriptions, and visualization results. The contents are organized as follows:

§A Prediction-space augmentation

§B Detailed network architecture

§C Detailed experimental setting

§D Parameter sensitivity analysis

§E Datasets details

§F Visualization results

A Prediction-space augmentation

A.1 Prediction-Space Augmentation

To construct semantic-level contrastive views, we perform augmentation directly in the prediction space rather than on the original input graph. Given a predicted graph $G_\theta = (X_\theta, A_\theta)$ generated by the diffusion model f_θ, we generate two stochastic views through a weak augmentation and a strong augmentation:

$$G_\theta^{(1)} = \mathcal{T}_w(G_\theta), \quad G_\theta^{(2)} = \mathcal{T}_s(G_\theta). \tag{15}$$

The weak augmentation introduces small stochastic perturbations to preserve the fine-grained semantic information of the predicted graph:

$$\mathcal{T}_w = \{\mathrm{NodeNoise}(\sigma_w), \mathrm{EdgeNoise}(\sigma_w)\}, \tag{16}$$

where $\sigma_w = 0.02$. Specifically, Gaussian noise is injected into the predicted node attributes and edge representations, respectively. Such a weak perturbation maintains the original semantic and structural patterns while providing stochastic diversity for contrastive learning.

The strong augmentation aims to improve semantic robustness by applying larger perturbations:

$$\mathcal{T}_s = \{\mathrm{NodeDrop}(p), \mathrm{NodeNoise}(\sigma_s), \mathrm{EdgeDrop}(p)\}, \tag{17}$$

where $\sigma_s = 0.05$ and $p = 0.5$. NodeDrop and EdgeDrop randomly mask node attributes and edge representations, respectively, encouraging the semantic encoder to capture invariant graph-level information beyond individual perturbations.

Different from conventional graph augmentation methods that modify the original observed graph, our strategy operates on the generated prediction G_θ, which naturally contains the semantic information learned by the diffusion model. The two augmented views are subsequently processed by the semantic encoder to obtain representations for contrastive learning. Through prediction-space augmentation, the model learns semantic consistency under stochastic variations without introducing additional sampling procedures.

B Detailed network architecture

B.1 Backbone denoising network

The denoising backbone network f_θ jointly models the interactions among node, edge, and global features through stacked Transformer blocks.

Category	Parameter	Ego-small	Enzymes	Planar	SBM	QM9	ZINC250k
Data	n (Max Nodes)	18	125	64	192	9	38
	d_x^{in} (Node Feat)	17	10	2	2	4	9
	d_a^{in} (Edge Feat)	2	2	2	2	2	2
Latent	d_x (Node Latent)	256	256	256	256	256	256
	d_a (Edge Latent)	64	64	64	64	128	64
	d_y (Global Latent)	64	64	64	64	64	64
FFN	dim_{ffx} (Node FFN)	256	256	256	256	256	256
	dim_{ffa} (Edge FFN)	64	64	64	64	256	128
Architecture	L (Layers)	4	8	8	8	8	9
	n_{head} (Heads)	8	8	8	8	8	8

Table 5: Hyperparameter configurations across different datasets.

Input Encoding and Higher-Order Structural Enhancement. In the initial stage, the network maps discrete input features into a continuous latent space through three independent linear projection layers. The node features $\mathbf{X} \in \mathbb{R}^{n \times d_x^{\text{in}}}$ and edge features $\mathbf{A} \in \mathbb{R}^{n \times n \times d_a^{\text{in}}}$ represent the perturbed graph structures at a specific diffusion time step t. Meanwhile, the global feature $\mathbf{y} \in \mathbb{R}^{d_y^{\text{in}}}$ is initialized as a time-step embedding that encodes the current denoising state t. These inputs are projected into latent dimensions d_x, d_a, and d_y, respectively. To enhance topology awareness, higher-order structural information is extracted from edge features using adjacency power operations. Physical consistency of the adjacency matrix is ensured through a symmetrization operation. All feature projections utilize the Rectified Linear Unit (ReLU) function to guarantee non-linear representational capacity.

Node-Edge-Global Co-evolutionary Attention Mechanism. The core of the network consists of L stacked co-evolutionary interaction layers, each integrating a multi-head self-attention module with n_{head} attention heads. This module transcends the limitations of traditional Transformers that process only sequential or node-level information. **Structure-Guided Attention Mechanism**: The attention scores between nodes across n_{head} independent heads depend not only on the dot product of their latent query and key vectors but are also modulated by the corresponding edge features. Utilizing Feature-wise Linear Modulation (FiLM) techniques, edge information is injected into the attention matrix as scaling and shifting factors. This allows pairwise interactions to adapt according to edge attributes. **Cross-Dimensional Information Feedback**: The global feature $\mathbf{y}$, which incorporates the encoding of the time step t, acts upon the node and edge features through a broadcasting mechanism. Simultaneously, the network aggregates local information, including mean, extrema, and standard deviation across node dimensions, to update the global vector $\mathbf{y}$. This enables a bidirectional information flow from local topology to macroscopic attributes.

Feature Evolution and Residual Connection Protocols. Following each interaction layer, the node, edge, and global vectors pass through independent Feed-Forward Networks (FFNs). The node path adopts a high-capacity design with an intermediate dimension dim_{ffx}, while the edge path utilizes a lightweight design with an intermediate dimension dim_{ffa} to balance computational overhead. Each path strictly implements residual connections and Layer Normalization. This architecture ensures gradient stability throughout the deep network during the denoising process.

Output Prediction Heads and Physical Constraints. At the terminal stage of the network, latent features are mapped back to their original dimensions via linear layers. To ensure that the generated molecular graphs conform to physical laws, specific post-processing logic is implemented: node features are assigned atomic category probabilities via a Softmax function, while edge features undergo mandatory symmetrization following Sigmoid activation. In specific branches dedicated to molecular tasks, the output layer performs a weighted fusion of different bond energy channels based on a preset scale, ultimately generating a continuous adjacency matrix.

To evaluate the effectiveness of the proposed architecture across diverse graph-structured tasks, we conducted extensive experiments on several benchmark datasets. Detailed hyperparameter configurations, including input dimensions and latent space settings for each layer, are summarized in Table 5.

B.2 Semantic Encoder

To extract multi-scale semantic information from the enhanced predictions, we implement a specialized semantic encoder .

Node Encoder (f_ψ) The node encoder consists of a 2-layer Multi-Layer Perceptron (MLP) with a hidden dimension of 256. It maps the augmented node features $X_\theta^{(i)}$ into a latent representation. To accommodate graphs with a variable number of nodes, we utilize Masked Global Average Pooling (M-GAP) to derive the graph-level node summary $\boldsymbol{h}_x^{(i)} \in \mathbb{R}^{256}$.

Edge Encoder (f_ϕ) The edge encoder is designed to capture global topological patterns. We flatten the augmented adjacency matrix $A_\theta^{(i)}$ into a 1D vector. For batches containing graphs smaller than the maximum node capacity, zero-padding is applied to ensure uniform input dimensions. This vector is processed by a 2-layer MLP (hidden dimension 256) to produce the structural embedding $\boldsymbol{h}_x^{(adj)} \in \mathbb{R}^{256}$.

Projection head (g_ξ) The decoder fuses the node-level and edge-level summaries via concatenation, resulting in a 512-dimensional composite representation. The decoder utilizes a two-layer MLP with Layer Normalization to stabilize representation learning and mitigate feature collapse, ultimately projecting the composite embedding into a 128-dimensional semantic space $\boldsymbol{z}^{(i)} \in \mathbb{R}^{128}$.

C Detailed experimental setting

C.1 Experimental environment

To ensure the reproducibility of our results, all experiments were conducted on a high-performance computing node equipped with a single NVIDIA GeForce RTX 3090 GPU (24GB VRAM). The software stack is built on PyTorch 1.13.1 and PyTorch Geometric (PyG) 2.3.1. For graph analysis and molecular property evaluation, we utilized NetworkX and RDKit.

Dataset	Batch size	Epoch	λ_g	λ_s	Time (h)	GPU Mem. (GB)
Planar	128	30000	0.05	0.05	5.0 ± 1.0	32.0 ± 3.0
SBM	32	30000	0.05	0.05	32.0 ± 3.0	55.0 ± 5.0
Ego-small	128	5000	0.05	0.05	1.0 ± 0.5	3.0 ± 1.0
Enzymes	32	5000	0.05	0.05	35.0 ± 3.0	15.0 ± 3.0
QM9	2048	1000	0.05	0.05	15.0 ± 2.0	16.0 ± 2.0
ZINC250k	512	1000	0.05	0.05	45.0 ± 5.0	55.0 ± 5.0

Table 6: Comparison of training hyperparameters of MCGD on different datasets.

C.2 Training and computational overhead

The model is optimized using AdamW with a weight decay of 10^{-12} and an EMA rate of 0.999.

Table 6 systematically summarizes and compares the training hyper-parameter settings and computational costs of MCGD across different datasets, aiming to illustrate the training configurations and resource requirements under various graph structures and scales.

The table lists multiple datasets, including synthetic or real-world structural graph datasets (Planar, SBM, Ego-small, Enzymes) as well as molecular graph datasets (QM9 and ZINC250k) , together

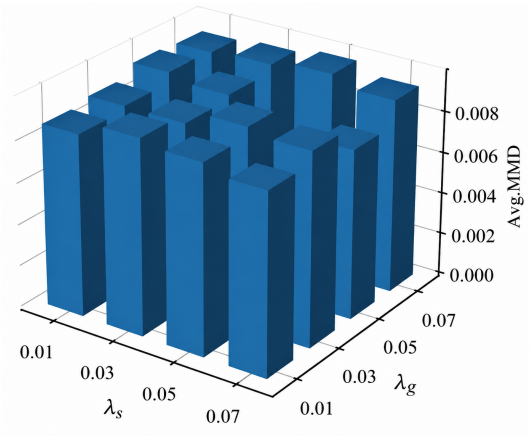


Figure 2: Parameter sensitivity analysis of λ_s and λ_g. The minimal Avg.MMD appears at $\lambda_s = 0.05$, $\lambda_g = 0.05$, and the model maintains stable performance within a wide hyperparameter range, verifying parameter insensitivity.

with their corresponding batch sizes and numbers of training epochs. These settings reflect different training strategies adopted for varying data scales and structural complexities.

The last two columns present the training time (in hours) and GPU memory consumption (in GB), reported as mean $\pm$ standard deviation, providing a direct comparison of computational costs across datasets. Overall, molecular datasets exhibit relatively manageable training times with larger batch sizes, whereas structural graph datasets (e.g., SBM) require more computational resources due to larger graph sizes or longer training schedules.

C.3 Inference and Sampling Details

During inference, we solve the reverse-time SDE using the Euler–Maruyama solver with $1,000$ discretization steps. To enhance structural fidelity, we adopt the Predictor-Corrector (PC) sampling scheme (Song et al., 2021b), where each predictor step is followed by a Langevin MCMC correction step for fine-grained structural refinement.

D Parameter Sensitivity Analysis

To investigate the sensitivity of the consistency weights, we evaluate the impact of two key hyperparameters, λ_g and λ_s, which control the contributions of graph-level and semantic-level consistency objectives, respectively. We vary both parameters within $0.01, 0.03, 0.05, 0.07 \times 0.01, 0.03, 0.05, 0.07$ and report the corresponding Avg.MMD scores in Figure 2.

As shown in Figure 2, the performance remains relatively stable across different combinations of λ_g and λ_s. The Avg.MMD values fluctuate within a narrow range, without significant degradation when either weight increases or decreases. This indicates that the proposed framework is robust to the selection of consistency weights and does not rely on careful parameter tuning.

The best performance is achieved with $\lambda_g = 0.05$ and $\lambda_s = 0.05$, where the model obtains the lowest Avg.MMD. However, nearby configurations yield comparable results, demonstrating that the improvement does not originate from a specific parameter setting but from the effectiveness of the proposed graph-level and semantic-level consistency regularization.

E Datasets Details

In this section, we present a detailed description of the datasets used to train the MCGD model, including their overall scale, the number of nodes in each molecule, and whether the molecules are synthetically generated.

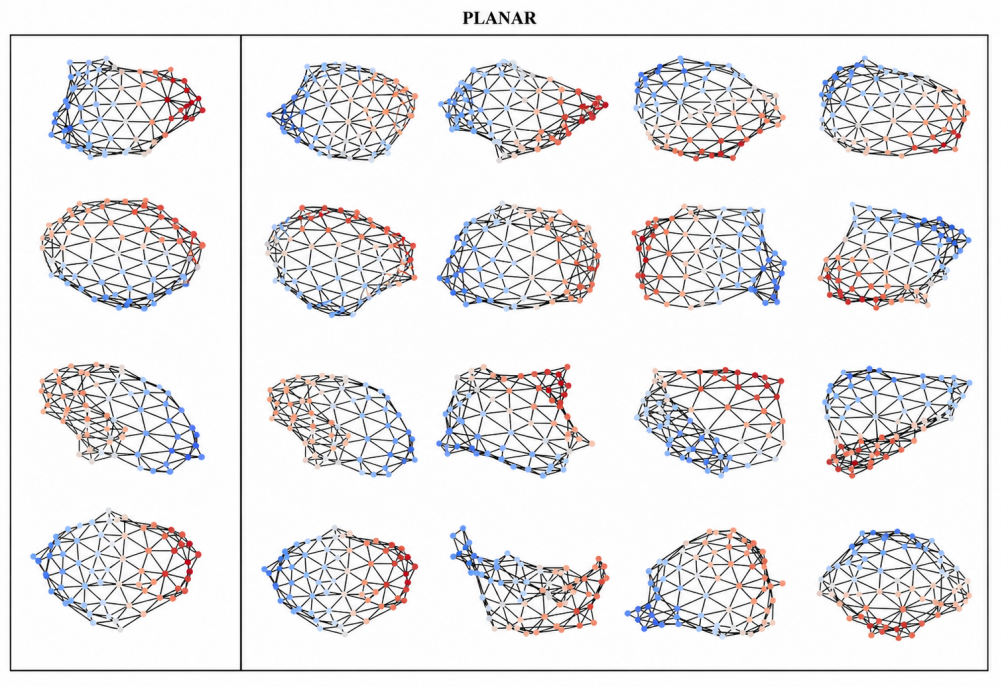


Figure 3: Visualization of graphs from the Planar dataset and the graphs generated by MCGD.

Planar The Planar Graph dataset contains 200 planar graphs with 64 nodes each, generated by randomly placing points on a 2D plane and applying Delaunay triangulation to form edges, ensuring planarity without crossings. Node features can include coordinates or identifiers, while edges correspond to triangulation connections. This dataset evaluates generative models under planar constraints and diverse topologies, providing reproducible benchmarks for latent space distribution and structural validity.

SBM The SBM (Stochastic Block Model) dataset consists of 200 synthetic graphs, each with 2–5 communities and 20–40 nodes per community. Intra-community edges are present with probability 0.3, inter-community edges with probability 0.05, and all edges are undirected and of a single type. Node features can optionally include additional labels or attributes. This dataset enables evaluation of graph generation models for community structure preservation, structural diversity, and latent space continuity.

Ego-small The Ego-small dataset comprises 200 small ego graphs extracted from the larger Citeseer citation network. Citeseer is a well-known academic citation graph where nodes represent papers and edges denote citation relationships. Each ego graph is constructed by selecting a central node along with its neighboring nodes, forming a local subgraph that typically contains between 4 and 18 nodes. Node types can be categorized according to paper topics or citation domains, while edges are undirected and of a single type. This dataset serves to evaluate a model's capability to generate realistic, sparse graphs with well-defined local structures, reflecting properties of real-world network topology.

Enzymes The Enzymes dataset contains 587 protein graphs, each representing the tertiary structure of a single enzyme. Nodes correspond to amino acid residues, classified into 3 types based on chemical properties or functional categories. Edges denote chemical or spatial interactions between residues and are undirected with a single type. Node counts range from 10 to 125. This dataset serves to assess model performance in generating biochemical graph structures and learning meaningful protein representations.

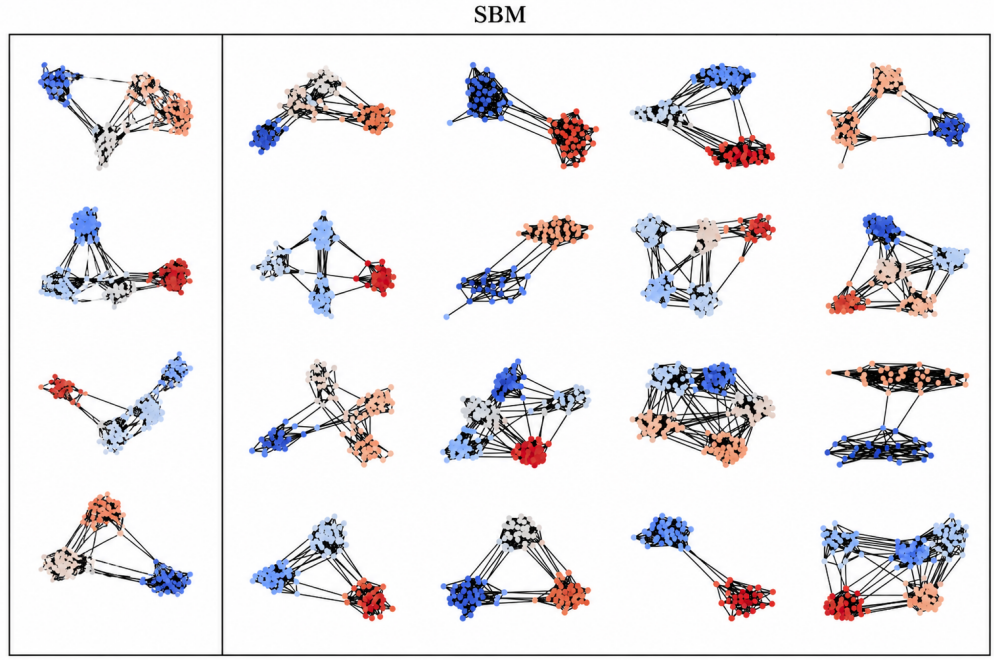


Figure 4: Visualization of graphs from the SBM dataset and the graphs generated by MCGD.

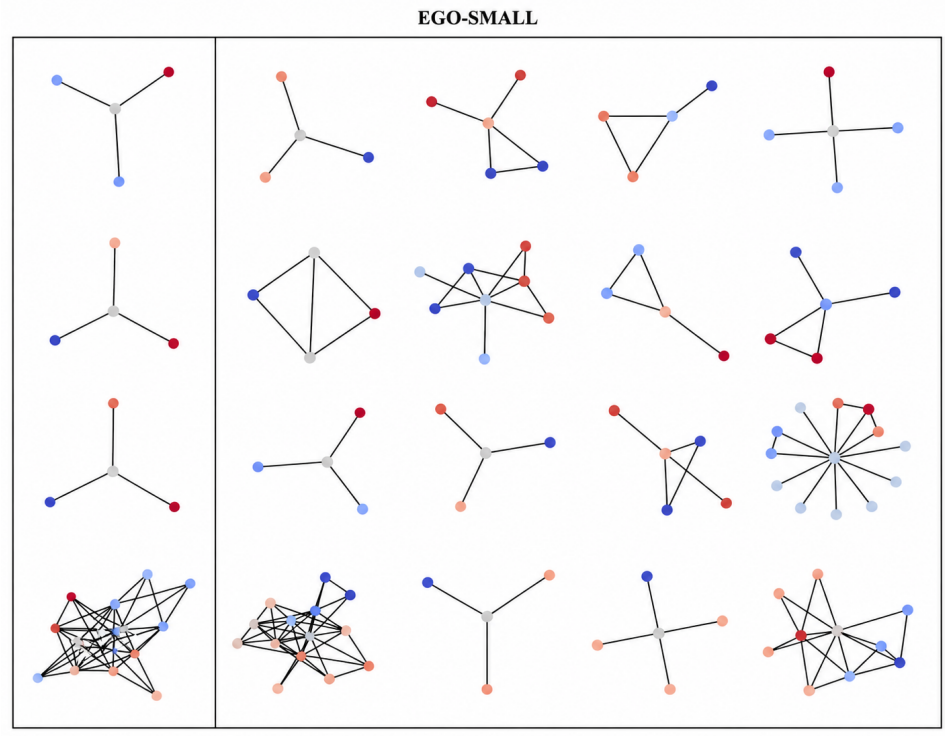


Figure 5: Visualization of graphs from the Ego small dataset and the graphs generated by MCGD.

QM9 We adopt the two-dimensional variant of the QM9 dataset, namely QM9-2D. It includes roughly 134,000 lightweight organic molecules consisting of atom species $\{\mathrm{C}, \mathrm{O}, \mathrm{N}, \mathrm{F}, \mathrm{H}\}$. Every molecule is formulated as a graph that maintains atomic connectivity as well as all bond classes (single,

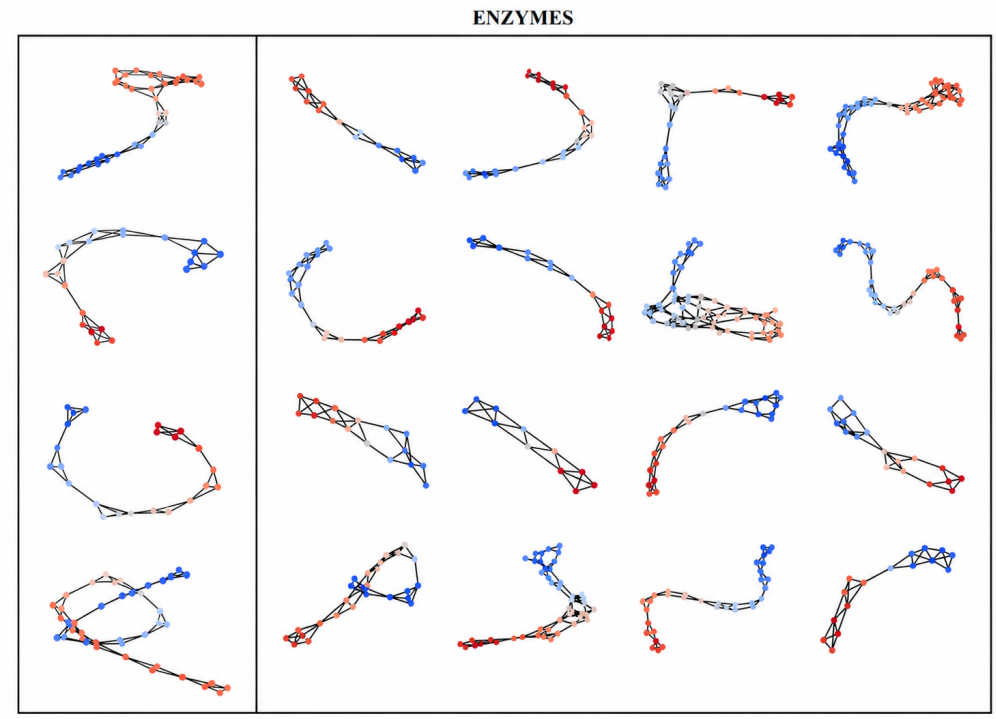


Figure 6: Visualization of graphs from the ENZYMES dataset and the graphs generated by MCGD.

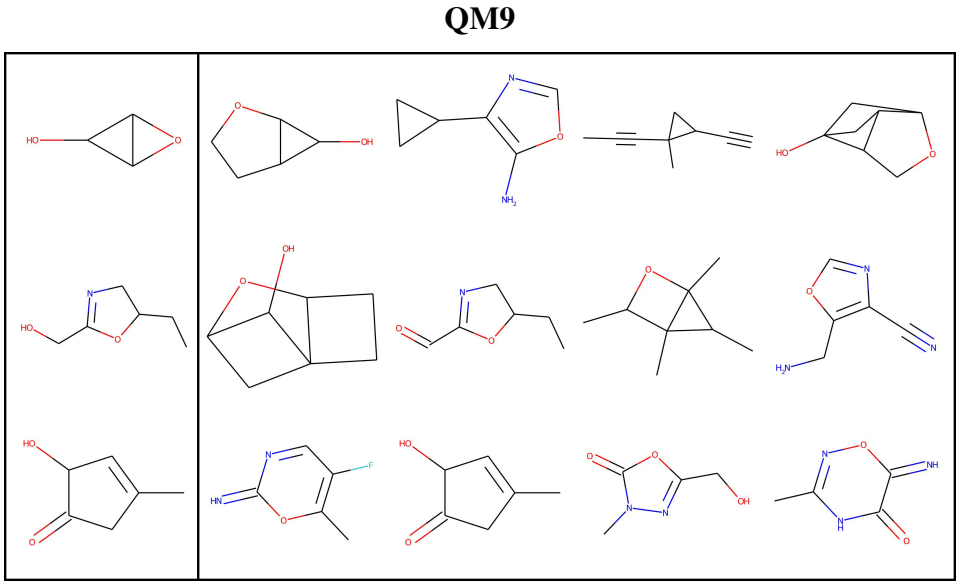


Figure 7: Visualization of graphs from the QM9 dataset and the graphs generated by MCGD.

double, triple, and aromatic bonds). Hydrogen atoms are deduced based on chemical valence rules. Node features encode atom types and discrete valence properties, whereas edge features characterize different bond categories. All molecules are transformed into adjacency matrices together with node feature matrices, with the maximum number of atoms limited to nine and four distinct bond types considered. This 2D representation highlights molecular topological structures, facilitating graph generative modeling at the graph level and removing extra geometric degrees of freedom.

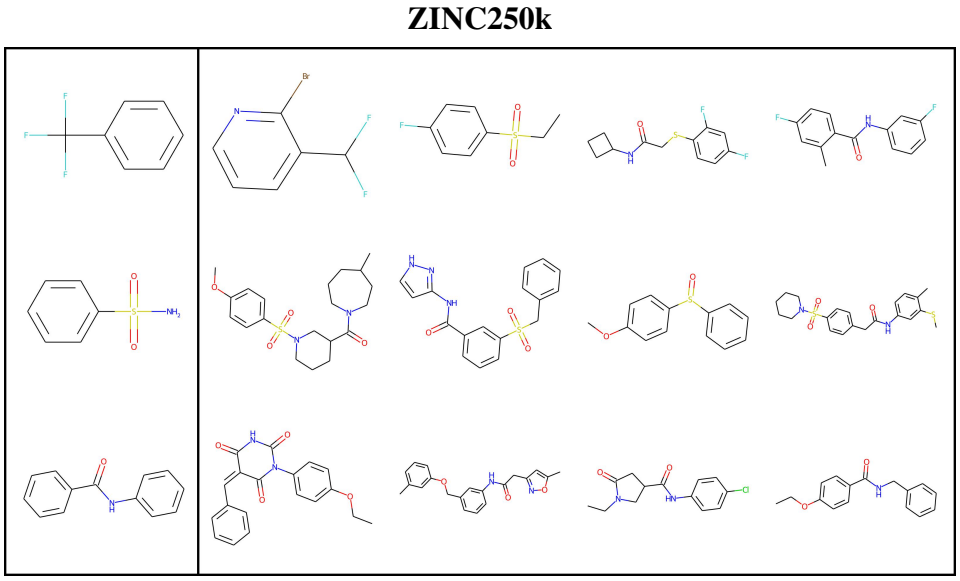


Figure 8: Visualization of graphs from the ZINC250k dataset and the graphs generated by MCGD.

ZINC250k The ZINC250k dataset consists of around 250,000 drug-like molecules, where each molecule possesses at most 38 atoms. The dominant element types cover C, N, O, F, S and Cl; a minor subset of samples additionally contain Br or P atoms. Individual molecules can be described via both SMILES and SELFIES string representations, and supported bond categories involve single, double and aromatic bonds. Benefiting from abundant structural variation, this dataset acts as a vital benchmark task for molecular generation and representation learning research.

F VISUALIZATION RESULTS

In this section, we present the molecular visualization results of our MCGD model on different datasets. For the visualization results, the left column shows the molecular graphs generated from the training set, while the right column presents the visualizations produced by our MCGD model.

F.1 VISUALIZATION OF GENERAL GRAPH

Figures 3, 4, 5, and 6 present the visualization results of the training sets and our MCGD model on the Planar, SBM, Ego-small and ENZYMES datasets.

F.2 VISUALIZATION OF MOLECULAR GRAPHS

Figures 7 and 8 show the visualization results for both the training sets and our MCGD model on the QM9 and ZINC250k molecular datasets.