GD-Mamba: Adaptive Cross-Domain Alternating Graph Diffusion Model with Selective State Space Model

Anonymous authors
Paper under double-blind review

Abstract

Graph generation is a critical task in deep learning, aimed at learning and modeling the structural distribution of graphs, with broad applications in areas such as social network analysis and molecular design. Current mainstream graph generation methods adopt diffusion models as the foundational architecture and typically apply them to either the discrete graph modality based on topological structures or the continuous latent space modality based on vector representations. Diffusion models for the graph modality focus on denoising discrete nodes and edges and lack unified control over the global structure, resulting in jump-like changes in the generation process. In contrast, diffusion models for the latent space modality perform denoising in a continuous space, resulting in gradual changes throughout the generation process. To incorporate global semantic guidance from the latent space modality into graph modality denoising, while injecting discrete priors from the graph modality into latent space denoising, we propose GD-Mamba, an Adaptive Cross-Domain Alternating Graph Diffusion Model with Selective State Space Model. GD-Mamba alternately performs denoising between the continuous and discrete modalities, namely the graph and latent space modalities. To stabilize the dual-modality generation process, we design a two-level Mamba structure (G-Mamba and S-Mamba) to align features across both modalities. Additionally, we introduce a multiscale wavelet positional encoding attention (MSW-PE) mechanism in the cross-domain mapping between both modalities. This mechanism effectively extracts the core structure of graphs by locating neighborhoods of nodes and edges at different frequency scales, improving the accuracy and robustness in the alternating denoising process. The experimental results demonstrate that the proposed GD-Mamba method improves generation quality on both generic graph datasets and molecular graph datasets.

Keywords Graph Generation, Diffusion Models, Deep Generative Models.

1 Introduction

Graph generation (Thompson et al., 2022; Tang et al., 2022) refers to modeling the distribution of graph data to enable the generation of new graph structures. It plays a critical role in domains such as drug design (Maziarz et al., 2021; Ahn et al., 2022), social network analysis (Du et al., 2021), and knowledge graph construction.

However, graph generation remains challenging due to the intrinsic duality of graph data, which encompasses two distinct representations: the discrete graph modality and the continuous latent space modality. The discrete graph modality explicitly represents topological structures via nodes and edges. Its discrete and sparse nature hinders long-range dependency modeling, often compromising overall connectivity and structural consistency. The continuous latent space modality encodes graphs into numerical vectors, effectively capturing global semantics but inevitably losing fine-grained structural details. Modeling the distributions of both modalities to achieve effective graph generation presents a considerable challenge. Earlier studies explored deep generative frameworks such as VAEs (Jin et al., 2018; Kipf & Welling, 2016), GANs (Wang et al., 2018; De Cao & Kipf, 2018), and normalizing flows (Zang & Wang, 2020; Madhawa et al., 2019; Luo et al., 2021). While

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.

these approaches demonstrate promising progress, they often struggle with training instability, mode collapse, or limited topological fidelity in complex graphs.

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020), which model data distributions through iterative forward noise addition and reverse denoising, have achieved remarkable results in image generation and natural language processing. This iterative paradigm has been increasingly adopted for graph generation (Niu et al., 2020; Gnaneshwar et al., 2022; Vignac et al., 2022). Existing works typically train these models separately on either the graph or latent space modality. Diffusion models operating on the graph modality are easily confined to local structures, lacking global semantic guidance. Latent-space diffusion models tend to amplify the loss of structural details during iterative denoising. To simultaneously capture global semantics and local structures in graph generation, we propose GD-Mamba, an adaptive cross-domain alternating Mamba diffusion model. Using a VAE-based cross-domain mapping module between the graph and latent spaces, GD-Mamba learns semantic and structural information across both modalities to achieve dual-modality iterative denoising. Our main contributions are summarized as follows:

- We propose GD-Mamba, an Adaptive Cross-Domain Alternating Graph Diffusion Model. By establishing a dual-modality alternating denoising mechanism, the model incorporates global semantic guidance from the latent space into graph denoising while integrating discrete structural priors into latent space denoising. This alternating generation approach improves the quality of graph generation.
- We design a two-level Mamba architecture (G-S Mamba) that aligns the discrete, jump-like topological transitions in the graph modality with continuous feature smoothing in the latent space, ensuring scale consistency and semantic unity throughout alternating denoising.
- We introduce a multiscale wavelet positional encoding attention (MSW-PE) mechanism into the Graph-Latent Mapping Module (GLMM), which leverages multi-frequency neighborhood context to preserve core graph topology against noise, thereby enhancing mapping accuracy and robustness.

2 Related Work

2.1 Diffusion model

The diffusion model is an emerging deep generative model that employs a diffusion process for data generation. It comprises two learnable Markov processes: a predefined forward process that progressively adds noise to the data, and a learnable reverse process that denoises the data to recover it. Mainstream diffusion models mainly adopt CNN-based UNet (Huang et al., 2020; Cao et al., 2023) or transformer (Vaswani et al., 2017; Parmar et al., 2018) denoising networks: DDPM (Ho et al., 2020) employs a UNet whose inductive bias benefits denoising, while DiT (Peebles & Xie, 2023), built upon stable diffusion (Rombach et al., 2022), introduces transformers into the image latent space and demonstrates favorable scalability. For graph generation, diffusion models follow two main directions: graph modality diffusion models and latent space modality diffusion models.

2.2 Graph diffusion model

Diffusion modeling for graph generation is particularly challenging in the discrete graph modality, where capturing complex node-edge relationships must be reconciled with the discreteness and sparsity of graphs. To address this, DiGress (Vignac et al., 2022) performs discrete diffusion directly on the graph modality by adding or removing edges and altering node types, better preserving discreteness and sparsity than continuous counterparts. However, the complexity of discrete space makes training diffusion models directly on the graph modality computationally expensive, motivating approaches that map graphs into a continuous latent space modality. For instance, NVDiff (Chen et al., 2022) employs a VAE to embed the discrete graph modality into a continuous latent space and models node vectors with a score-based diffusion model (Song et al., 2021), substantially reducing the diffusion dimensionality and accelerating sampling. While this latent space modality diffusion method is effective, its performance heavily depends on the quality of the VAE training, particularly the alignment between the prior and posterior distributions and the graph's reconstruction

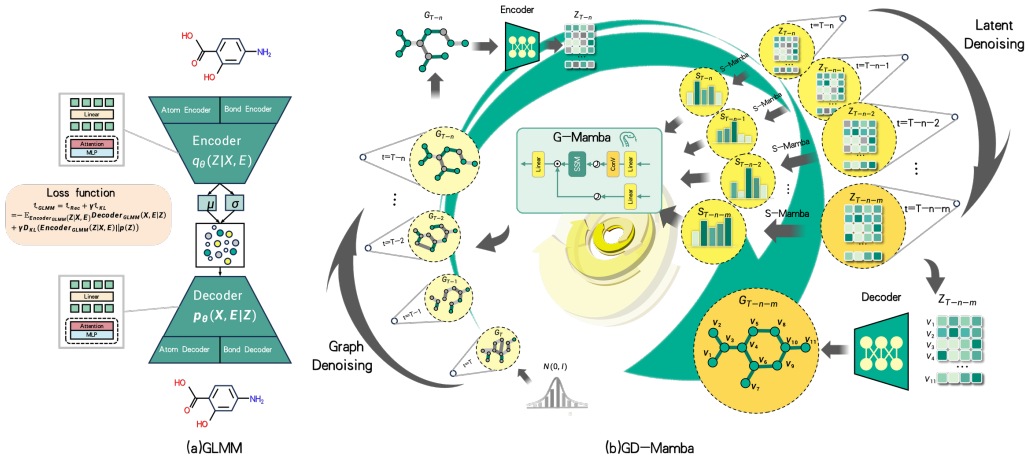


Figure 1: Overall framework of GD-Mamba. (a) Encoder and decoder structure of the Graph-Latent mapping module. (b) Reverse denoising process of the GD-Mamba.

ability. Structural misalignment during the reconstruction process tends to intensify over the course of diffusion, thereby affecting the semantic and structural consistency of the generated graphs.

2.3 State space models

State space models (SSMs) (Wang & Xue, 2023), which hybridize recurrent and convolutional computation, have shown strong capacity for long-sequence modeling, with variants such as S4 (Gu et al., 2021) and Hyena (Nguyen et al., 2023) progressively improving efficiency and context capture. Mamba (Gu & Dao, 2023) further introduces a selective SSM that integrates time-varying parameters and a hardware-aware scanning algorithm to reduce computational overhead. SSMs further extend beyond sequential data: Vision Mamba (Zhu et al., 2024) introduces a bidirectional SSM that captures global context with subquadratic complexity and linear memory, outperforming ViT (Yuan et al., 2021) in efficiency while maintaining strong modeling capability.

As traditional graph transformers (Ying et al., 2021) scale poorly on large graphs due to the quadratic complexity of self-attention, recent studies extend SSMs to graph-structured data. Graph-Mamba (Wang et al., 2024) first introduces SSMs into graph networks, proposing a selective SSM module that captures long-range dependencies through node prioritization and adaptive context selection. It outperforms existing methods on long-range graph prediction tasks while substantially reducing GPU memory consumption. This opens novel pathways for SSMs in graph data modeling.

3 Method

In this section, we present GD-Mamba, an adaptive cross-domain alternating graph diffusion framework that synergistically couples the discrete graph modality with the continuous latent space modality. As illustrated in Figure 1, GD-Mamba comprises three key components: (1) an overall dual-modality alternating denoising schedule that coordinates diffusion steps between discrete and latent spaces; (2) a two-level Mamba architecture (G-S Mamba, Section 3.1) that aligns discrete topological transitions with continuous feature smoothing; and (3) a multiscale wavelet positional encoding mechanism (MSW-PE) integrated into the Graph-Latent Mapping Module (GLMM, Section 3.2) to preserve structural fidelity during cross-domain projection. Part (a) of Figure 1 depicts the encoder–decoder mapping module (GLMM), and Part (b) outlines the dual-modality alternating denoising flow. Below, we first formalize the alternating denoising procedure before detailing each module.

Let G and Z represent the graph and latent space modality representations, respectively; t denotes the current denoising time step; and T is the maximum time step in the GD-Mamba alternating denoising algorithm. Initially, a sample of the graph modality G_T is drawn from a Gaussian distribution, and

n steps of denoising are performed to obtain $G_{(T-n)}$. Next, the encoder of the mapping module maps $G_{(T-n)}$ to the latent space modality $Z_{(T-n)}$. The process then transitions to the latent space modality denoising stage, where m steps of denoising are applied to $Z_{(T-n)}$, resulting in $Z_{(T-n-m)}$. Finally, the decoder transforms $Z_{(T-n-m)}$ back into the graph modality $G_{(T-n-m)}$. Repeating these steps realizes alternating denoising between the two modalities.

3.1 Two-level Mamba Module for coordinated dual-modal denoising

The two modalities denoise in fundamentally different ways: graph modality denoising reconstructs topological connections and node attributes, where changes occur as discrete jumps between entities, whereas latent space denoising progressively smooths latent features, emphasizing gradual changes. Once graph modality denoising completes, its discrete features are mapped to the latent space for further denoising, where these mechanistic differences create a scale-level disparity between the two processes.

Since both the discrete, jump-like denoising of the graph modality and the gradual denoising of the latent space modality can be viewed as a temporal evolution where noise diminishes and structure emerges, Mamba—which excels at modeling complex dependencies in long sequences—is well suited to both. To address the scale-level differences between the two modalities, we propose two-level Mamba modules (G-Mamba and S-Mamba), as shown in Figure 1. The G-Mamba module is responsible for denoising the graph modality, whereas the S-Mamba module (i.e., the latent space denoising network SSDM, detailed in Section 3.3) focuses on denoising the latent space modality. The multistep continuous features extracted by the S-Mamba module from the latent space modality serve as conditional information to guide the G-Mamba module in denoising the graph modality. G-Mamba itself adopts a standard Mamba block—linear projection, convolution, and a selective SSM combined through a gated branch (inset of Figure 1(b))—as our contribution lies in the two-level coordination rather than in the block design. Through the coordinated training of the G-S Mamba modules, the discrete, jump-like changes in the graph modality are harmonized with the gradual changes in the latent space modality, ensuring smooth transitions and semantic consistency during the dual-modality denoising process.

The training of the latent space diffusion model is adapted for S-Mamba, which serves as the latent space denoising network. First, noise is added to the latent space modality Z_0, generating a noisy state Z_t. Then, through the reverse process of the diffusion model, Z_t is progressively restored to earlier states, Z_{t-n}, including all intermediate states between Z_t and Z_{t-n}. This process can be described by the following equation:

$$Z_{t-k} = DDPM_{\text{reverse}}(Z_{t-k+1}, \text{SSDM}(Z_{t-k+1}, t-k+1)) \tag{1}$$

In this context, $DDPM_{reverse}$ denotes the reverse process of the diffusion model. At each denoising step, S-Mamba extracts features S_t to S_{t-n} that reflect the continuous changes in the latent space modality and serve as conditional information for training G-Mamba.

$$S_{t-k} = \text{SSDM}(Z_{t-k}) \tag{2}$$

G-Mamba is then trained to learn the discrete structures of the graph modality. First, Z_t and Z_{t-n} are mapped back to the graph modality as G_t and G_{t-n} via the mapping module. Then, the continuous features $S_t, S_{t-1}, \ldots, S_{t-n}$, extracted by S-Mamba, are combined as conditional information to guide the G-Mamba module in completing the denoising process for the graph modality. The loss function is defined as:

$$L_{\text{G-Mamba}} = \left|\text{G-Mamba}\left(G_t, \text{Cond}\left(S_t, S_{t-1}, \ldots, S_{t-n}\right)\right) - G_{t-n}\right|^2 \tag{3}$$

After training, G-S Mamba is applied during alternating denoising to achieve scale alignment between the two modalities, as shown in Figure 1(b). The S-Mamba extracts latent features S_{T-n} to S_{T-n-m} from the latent space during the denoising phase of Z_{T-n} to Z_{T-n-m} and uses these as conditional inputs for G-Mamba.

3.2 Cross-domain mapping module (GLMM)

To perform modality alternation, we design a cross-domain mapping module, GLMM, which uses a transformer as its backbone, as illustrated in Figure 1(a). GLMM consists of an encoder q_θ mapping the graph modality to the latent space modality and a decoder p_θ mapping it back.

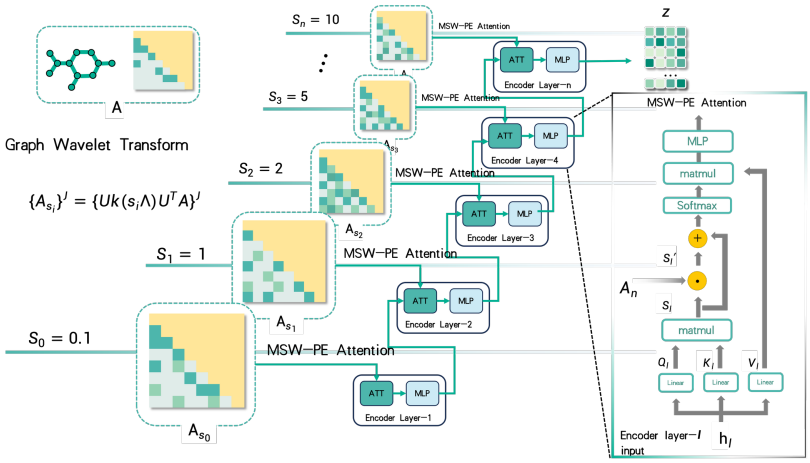


Figure 2: MSW-PE Module

In the dual-modality alternating denoising process, noise interference can cause the discrete structure of the graph modality to shift. When the GLMM encodes a noisy graph, the noise disrupts the relational structure between nodes, leading to latent space modality node features that carry noise information and fail to accurately reflect the core structure of the graph. This mismatch may result in the loss of information in the latent space, thereby weakening the subsequent denoising effectiveness of the latent space modality. To preserve the semantic integrity of the latent space after encoding a noisy graph, we introduce a multiscale wavelet positional encoding attention (MSW-PE attention) mechanism into the GLMM. It applies multithreshold wavelet transforms to the adjacency matrix to generate multiscale adjacency matrices, which serve as positional encodings for the multilevel GLMM encoder, enabling GLMM to locate node and edge neighborhood associations across frequency scales and thus preserve the core structure of the graph during alternating denoising.

Positional encodings help the encoder capture graph topology by contextualizing node and edge neighborhoods, but traditional adjacency-matrix positional encodings are easily disrupted by noise when transforming between modalities at different noise levels—especially when switching between local discrete noise in the graph modality and continuous noise in the latent space modality. In the discrete graph modality, noise typically manifests as high-frequency signals, whereas the global structure or stable local connections of the graph are more reflected in low-frequency components. Multiscale graph wavelet transforms separate these frequency bands, letting the mapping module capture core topology and stable features in the low-frequency range while identifying noise and local abnormal connections in the high-frequency range. Therefore, MSW-PE attention locates node and edge neighborhood context across frequency scales, extracting the core graph structure while suppressing noise and improving mapping accuracy.

Figure 2 details the MSW-PE mechanism. First, the adjacency matrix $\mathbf{A}$ of the input graph data undergoes a graph wavelet transform. By computing the graph's Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and performing spectral decomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, the eigenvector matrix $\mathbf{U}$ and eigenvalue matrix $\mathbf{\Lambda}$ are obtained. Then, using a mother wavelet function $k(s)$, multiscale wavelet basis functions $\psi_s = \mathbf{U}k(s\mathbf{\Lambda})\mathbf{U}^T$ are constructed, generating multiscale adjacency matrices $\mathbf{A}_s = \mathbf{U}k(s\mathbf{\Lambda})\mathbf{U}^T\mathbf{A}$, where the scale parameter s controls the spectral bandwidth of the graph signal. These multiscale adjacency matrices are injected as positional encodings into different layers of the encoder within the mapping module.

For each encoder layer, the input feature $\mathbf{H}$ is mapped to $\mathbf{Q}_l \in \mathbb{R}^{N\times W}$, $\mathbf{K}_l \in \mathbb{R}^{N\times W}$, and $\mathbf{V}_l \in \mathbb{R}^{N\times W}$ through three learnable weight matrices $\mathbf{W}_l^Q$, $\mathbf{W}_l^K$, and $\mathbf{W}_l^V$, where l denotes the index of the encoder layer.

$$\mathbf{Q}_l = \mathbf{H}\mathbf{W}_l^Q \quad \mathbf{K}_l = \mathbf{H}\mathbf{W}_l^K \quad \mathbf{V}_l = \mathbf{H}\mathbf{W}_l^V \tag{4}$$

The attention score matrix is then computed as $\mathbf{S}_l = \mathbf{Q}_l\mathbf{K}_l^T \in \mathbb{R}^{N\times N}$. Next, the multiscale adjacency matrix $\mathbf{A}_s$ and the attention score matrix $\mathbf{S}_l$ undergo the Hadamard (elementwise) product,

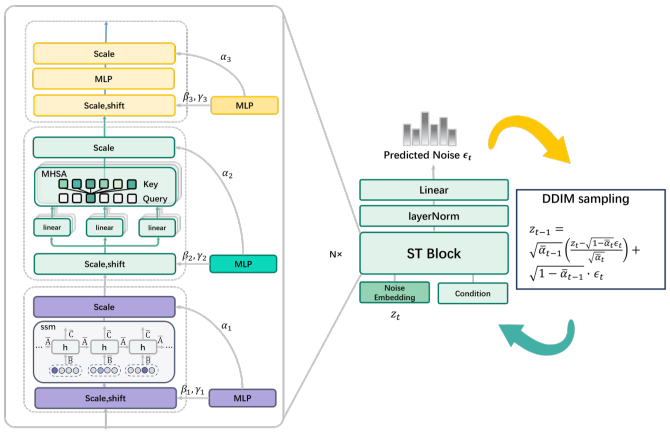


Figure 3: Structure of SSDM

and the result is added back to $\mathbf{S}_l$ via a residual connection, updating the attention score matrix as $\mathbf{S}'_l = \mathbf{S}_l + \mathbf{S}_l \odot (y_l * \mathbf{A}_s)$, where y_l is a learnable vector. The result of the MSW-PE attention computation is as follows:

$$O_l = \text{softmax}\left(\frac{S'_l}{W}\right) V_l \tag{5}$$

After passing through the MSW-PE attention mechanism, the output is further processed by a multilayer perceptron (MLP) to enhance the feature representation capabilities.

The cross-domain mapping module between the graph and latent space is trained via both reconstruction loss and KL divergence loss, with the loss function defined as follows:

$$\mathcal{L}_{\text{GLMM}} = \mathcal{L}_{\text{Rec}} + \gamma \mathcal{L}_{\text{KL}} = -\mathbb{E}_{q_\theta(\mathbf{Z}|\mathbf{X},\mathbf{E})}\, p_\theta(\mathbf{X}, \mathbf{E} \mid \mathbf{Z}) + \gamma D_{\text{KL}}\left(q_\theta(\mathbf{Z} \mid \mathbf{X}, \mathbf{E}) \| p(\mathbf{Z})\right) \tag{6}$$

where γ is the weight of the KL divergence loss and the KL loss is used to align the latent space $\mathbf{Z}$ with the standard Gaussian distribution $p(\mathbf{Z})$.

3.3 SSDM: Selective SSM Diffusion Model

Small changes in the discrete graph modality often require multiple steps of gradual adjustment in the continuous space. To capture these progressive changes, S-Mamba is instantiated as the selective SSM diffusion model (SSDM).

The SSDM model is based on the diffusion transformer (DiT) architecture combined with state space models (SSMs), leveraging the feature extraction capabilities of the transformer to handle noise distributions while incorporating the SSM structure to achieve multilevel denoising control. SSMs, as adaptive selection mechanisms, can adjust parameters in real-time on the basis of the input data, filter out irrelevant information, and capture gradual continuous changes in the latent space modality during the denoising process. Through adaptive updates, SSMs can track and retain these continuous changes, using them as guidance for G-Mamba in denoising the discrete graph modality.

The architecture of SSDM is shown in Figure 3, and its structure can be described as follows (latent-space sampling at inference follows the DDIM update shown in the same figure):

$$S = \text{STBlock}(Z_t, c), \quad \epsilon'_t = \text{LinearLayer}(\text{LayerNorm}(S)) \tag{7}$$

where Z_t is the noisy latent space vector, and c is the input condition. After processing through N layers of the STBlock, the intermediate feature S is obtained, and finally, the predicted noise ϵ'_t is generated through the LinearLayer.

The specific details of the STBlock are shown in Figure 3, and it consists of three modules: a selective SSM, a multihead self-attention mechanism, and a feedforward neural network. Each module is equipped with adaptive layer normalization (AdaLN block) to increase the flexibility and

effectiveness of feature processing. In the STBlock, the input condition c is passed through three condition encoders (MLPs) to generate three sets of scaling and shifting parameters, which are used to adaptively normalize the intermediate features in each module of the STBlock. The formula for adaptive normalization is as follows:

$$\text{AdaLN block}(H, c) = \gamma_i(c) \odot H + \beta_i(c) \tag{8}$$

where H represents the input feature for each module, whereas $\gamma_i(c)$ and $\beta_i(c)$ are the scaling and shifting parameters generated by the corresponding condition encoder MLP_i on the basis of condition c. To further increase the flexibility of feature representation, an adaptive normalization gating mechanism is introduced in the STBlock. This mechanism regulates the normalized output through gating parameters $\alpha_i(c)$ generated by the condition encoder MLP_i and is defined as follows:

$$\text{AdaLN}_{gate}(H, c) = \alpha_i(c) \odot \text{AdaLN block}(H, c) \tag{9}$$

During the training process of the latent space modality diffusion model, the noisy feature Z_t is used as the model input, passing through the N stacked STBlocks of SSDM. The final output is the predicted noise ϵ'_t, and the loss between the predicted noise ϵ'_t and the true noise ϵ_t is calculated via the MSE loss function. The loss function is defined as follows:

$$\mathcal{L}_{\text{SSDM}} = \|\epsilon_t - P_{SSDM}(Z_t, t)\|^2 = \|\epsilon_t - \epsilon'_t\|^2 \tag{10}$$

4 Experiments

4.1 General Graph Generation

The model was first evaluated on four general graph datasets: Ego-small, Community-small, Enzymes, and Grid. Three types of graph features were used to evaluate the MMD distances between the generated graphs and the test graphs: degrees (Deg.), the clustering coefficients (Clus.), and orbit counts (Orbit). Dataset and metric details are given in Appendix A.

The baselines were GraphRNN (You et al., 2018), GraphAF (Shi et al., 2020), GraphDF (Luo et al., 2021), GraphVAE (Simonovsky & Komodakis, 2018), EDP-GNN (Niu et al., 2020), GDSS (Jo et al., 2022), and DiGress (Vignac et al., 2022).

4.2 Molecular Graph Generation

The model was trained and evaluated on two molecular datasets: QM9 (Ramakrishnan et al., 2014) and ZINC250k (Irwin et al., 2012), using the standard RDKit preprocessing described in Appendix A. The molecules generated by our model were then evaluated via the Fréchet ChemNet Distance (FCD) (Preuer et al., 2018), the neighborhood subgraph pairwise distance kernel (NSPDK) MMD (Costa & De Grave, 2010), *validity w/o correction*, *uniqueness*, and *novelty*; Appendix A provides the full definitions of all metrics.

The baselines covered graph-modality diffusion models (GDSS (Jo et al., 2022), DiGress (Vignac et al., 2022), and EDP-GNN (Niu et al., 2020)), a latent-space-modality diffusion model (NVDiff (Chen et al., 2022)), end-to-end generative models (GraphAF (Shi et al., 2020) and GraphDF (Luo et al., 2021)), a flow-based model (MoFlow (Zang & Wang, 2020)), and an energy-based model (GraphEBM (Liu et al., 2021)). Results of MoFlow, GraphAF, GraphDF, EDP-GNN, GraphEBM, and GDSS were taken from Jo et al. (2022), and the validity, NSPDK, and FCD of DiGress from Jo et al. (2024).

4.3 Experimental Results and Analysis

General Graph Generation: Table 1 presents the performance metrics achieved on the general graph datasets. GD-Mamba attained the lowest average MMD on three of the four datasets (Ego-small, Community-small, and Enzymes) and the second-lowest on Grid, surpassing both non-diffusion and diffusion-based baselines in most settings. This indicates that GD-Mamba effectively captures both

Table 1: Generic graph generation performance.

	Ego - small			Community - small			Enzymes			Grid		
	$4 \leq \|V\| \leq 18$			$12 \leq \|V\| \leq 20$			$10 \leq \|V\| \leq 125$			$100 \leq \|V\| \leq 400$		
Method	Deg.	Clus.	Orb.	Deg.	Clus.	Orb.	Deg.	Clus.	Orb.	Deg.	Clus.	Orb.
GraphVAE	0.130	0.170	0.050	0.350	0.980	0.540	1.369	0.629	0.191	1.619	**0.000**	0.919
GraphRNN	0.090	0.220	0.003	0.080	0.120	0.400	**0.017**	0.062	0.046	**0.064**	0.043	**0.021**
EDP-GNN	0.052	0.093	0.007	0.053	0.144	0.026	0.023	0.268	0.082	0.455	0.238	0.328
GraphAF	0.034	0.110	**0.001**	0.181	0.205	0.023	1.669	1.283	0.266	-	-	-
GraphDF	0.041	0.135	0.010	0.064	0.125	0.037	1.503	1.061	0.202	-	-	-
GDSS	0.021	0.024	0.007	0.045	0.086	0.007	0.026	0.061	0.009	0.111	0.005	0.070
DiGress	0.025	0.021	0.008	0.051	0.083	0.009	0.030	0.060	0.011	0.189	0.003	0.070
GD-Mamba (ours)	**0.011**	**0.015**	0.007	**0.009**	**0.014**	**0.004**	0.018	**0.035**	**0.007**	0.108	**0.000**	0.048

Table 2: Molecular graph generation performance.

	QM9					ZINC250k				
Method	Val.↑	NSPDK↓	FCD↓	Uniq.↑	Nov.↑	Val.↑	NSPDK↓	FCD↓	Uniq.↑	Nov.↑
EDP-GNN	47.52	0.005	2.68	**99.25**	86.58	82.97	0.049	16.74	99.79	**100**
MoFlow	91.36	0.017	4.47	98.65	94.72	63.11	0.046	20.93	**99.99**	**100**
GraphAF	74.43	0.020	5.63	88.64	86.59	68.47	0.044	16.02	98.64	**100**
GraphDF	93.88	0.064	10.93	98.58	**98.54**	90.61	0.177	33.55	99.63	**100**
GraphEBM	8.22	0.030	6.14	97.90	97.01	5.29	0.212	35.47	98.79	**100**
GDSS	95.72	0.003	2.9	98.46	86.27	97.01	0.019	14.66	99.64	**100**
DiGress	98.19	**0.0003**	**0.095**	96.34	35.46	94.99	**0.002**	**3.48**	99.97	**100**
NVDiff	95.79	0.002	1.13	96.53	-	85.63	0.06	4.02	**99.99**	-
Latent-only (ours)	96.21	0.004	1.43	97.74	72.32	92.32	0.042	4.23	99.43	**100**
GD-Mamba (ours)	**99.52**	0.002	1.04	93.52	53.32	**98.94**	0.016	3.64	98.21	**100**

the global semantic information of the latent space modality and the structural distribution of the graph modality, excelling on graphs with up to 125 nodes while remaining competitive on the larger Grid graphs.

Unconditional Molecular Graph Generation: Table 2 presents the generation performance metrics attained by the GD-Mamba method on the molecular graph datasets (QM9 and ZINC250k), with the mean of three samples used as the final result. Without valence correction, GD-Mamba attained the highest molecular validity on both QM9 (99.52%) and ZINC250k (98.94%), surpassing the leading graph-modality diffusion models (DiGress and GDSS) and the latent-space-modality diffusion model (NVDiff). The high validity scores suggest that GD-Mamba, through the encoder–decoder structure of GLMM, learns chemically plausible valence rules. Moreover, GD-Mamba attained the second-best NSPDK MMD and FCD on both datasets, behind only DiGress, demonstrating that the GD-Mamba model not only learns the global semantic information of the latent space modality but also captures the structural distribution information of the graph modality, effectively representing molecular structures and chemical properties. The novelty metric on QM9 remains debatable: since QM9 encompasses nearly all molecules satisfying its construction criteria, generating novel molecules outside this set does not necessarily reflect success in capturing the underlying data distribution. Novelty is still reported for completeness.

The results produced by the latent-space-modality diffusion model from Section 3.3 were also evaluated (listed as Latent-only in Table 2). Without the guidance of the graph modality structure, the single-latent-space-modality diffusion model performed slightly worse than GD-Mamba in capturing the graph structure distribution, with lower validity and higher NSPDK MMD without valence correction. This demonstrates the advantage of the proposed dual-modality alternating denoising for capturing graph structure distributions.

4.4 Ablation Study

The effect of different position encodings on GLMM reconstruction was first examined, comparing three strategies: MSW-PE, adjacency matrix-based encoding, and no position encoding, as the number of active training epochs increases. As shown in Figure 4, MSW-PE consistently attained

the highest reconstruction accuracy throughout training, followed by adjacency-based encoding and then no position encoding, confirming the role of multiscale position encoding in improving the performance and stability of GLMM.

The impact of varying step distributions between G-Mamba and S-Mamba was next evaluated on the model's performance, measured by Validity w/o correction, using the ZINC250k dataset. The total number of denoising steps was fixed at 1000, distributed across 10 cycles, with each cycle consisting of 100 steps. In each cycle, seven different distributions were examined, where S-Mamba steps decreased progressively from 100 to 40, while G-Mamba steps increased accordingly. The results, shown in Table 3, reveal that the most favorable balance occurred when G-Mamba steps are around 30 and S-Mamba steps are 70, yielding the highest performance of 97.94%. As this ablation targets relative trends across step allocations, its absolute values differ slightly from the three-sample means in Table 2. These results demonstrate that alternating denoising, combining both discrete and continuous steps, enhances the model's ability to learn distributions and improves its denoising performance. As G-Mamba steps increased beyond this point and S-Mamba steps decreased, performance declined. This suggests that when G-Mamba steps dominate, the excessive discrete steps introduce instability and information loss, impeding the coordination between the two modalities.

5 Conclusion

We propose GD-Mamba, an Adaptive Cross-Domain Alternating Graph Diffusion Model with Selective State Space Model. It performs alternating denoising between the two modalities: global semantic features from the latent space modality guide graph modality denoising to ensure semantic consistency, while discrete priors from the graph modality provide structural guidance for latent space modality denoising to preserve structural rationality. To unify their denoising scales, we design a two-level Mamba structure (G-Mamba and S-Mamba) that aligns continuous and discrete features throughout the alternating denoising process. We further introduce a multiscale wavelet positional encoding attention mechanism (MSW-PE attention) into the cross-domain mapping module (GLMM), which locates neighborhood associations of nodes and edges across frequency scales, extracting the core graph structure and enhancing the accuracy and robustness of cross-domain mapping. In experimental validation, GD-Mamba attained the lowest average MMD on most general graph datasets and the highest molecular validity on both QM9 (99.52%) and ZINC250k (98.94%) without valence correction, while remaining competitive on distribution metrics, further demonstrating its modeling capability for graph data semantics and structure. The optimal ratio between the G-Mamba and S-Mamba denoising steps currently requires manual tuning, and a deviation from it degrades generation quality. Future work will investigate an adaptive framework for allocating the denoising steps across modalities, and will apply GD-Mamba to more complex graph data and conditional generation tasks such as target protein generation.

AI use statement

Generative AI tools were used solely to assist with language editing, text polishing, and terminology consistency checks of this manuscript. All conceptual frameworks, mathematical formulations, experimental methodologies, analyses, and conclusions were independently developed and verified by the authors. The authors take full responsibility for the content and integrity of this paper.

References

Sungsoo Ahn, Binghong Chen, Tianzhe Wang, and Le Song. Spanning tree-based graph generation for molecules. In *International Conference on Learning Representations*, 2022.

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *Computer Vision – ECCV 2022 Workshops*, pp. 205–218. Springer, 2023.

Xiaohui Chen, Yukun Li, Aonan Zhang, and Li-ping Liu. Nvdiff: Graph generation through the diffusion of node vectors. *arXiv preprint arXiv:2211.10794*, 2022.

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 255–262, 2010.

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

Yuanqi Du, Shiyu Wang, Xiaojie Guo, Hengning Cao, Shujie Hu, Junji Jiang, Aishwarya Varala, Abhinav Angirekula, and Liang Zhao. Graphgt: Machine learning datasets for graph generation and transformation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Dwaraknath Gnaneshwar, Bharath Ramsundar, Dhairya Gandhi, Rachel Kurchin, and Venkatasubramanian Viswanathan. Score-based generative models for molecule generation. *arXiv preprint arXiv:2203.04698*, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 1055–1059. IEEE, 2020.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7): 1757–1768, 2012.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International conference on machine learning*, pp. 10362–10383. PMLR, 2022.

Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. *arXiv preprint arXiv:2302.03596*, 2024.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. GraphEBM: Molecular graph generation with energy-based models. In *Energy Based Models Workshop, International Conference on Learning Representations (ICLR)*, 2021.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation. In *International conference on machine learning*, pp. 7192–7203. PMLR, 2021.

Kaushalya Madhawa, Katushiko Ishiguro, Kosuke Nakago, and Motoki Abe. Graphnvp: An invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600*, 2019.

K Maziarz, H Jackson-Flux, P Cameron, F Sirockin, N Schneider, N Stiefl, and M Brockschmidt. Learning to extend molecular scaffolds with structural motifs. *arXiv preprint arXiv:2103.03864v1*, 2021.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, et al. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. *arXiv preprint arXiv:2306.15794*, 2023.

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

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pp. 4055–4064. PMLR, 2018.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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(suppl_1):D431–D433, 2004.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: A flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pp. 412–422. Springer, 2018.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

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

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

Mingyue Tang, Carl Yang, and Pan Li. Graph auto-encoder via neighborhood wasserstein reconstruction. *arXiv preprint arXiv:2202.09025*, 2022.

Rylee Thompson, Boris Knyazev, Elahe Ghalebi, Jungtaek Kim, and Graham W Taylor. On evaluation metrics for graph generative models. *arXiv preprint arXiv:2201.09871*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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.

Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024.

Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Shida Wang and Beichen Xue. State-space models with layer-wise nonlinearity are universal approximators with exponential decaying memory. *Advances in Neural Information Processing Systems*, 36, 2023.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pp. 5708–5717. PMLR, 2018.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 558–567, 2021.

Chengxi Zang and Fei Wang. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 617–626, 2020.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

A Datasets, Metrics, and Preprocessing

(1) **Ego-small** (Sen et al., 2008): 200 small graphs extracted from the Citeseer network; (2) **Community-small**: 100 randomly generated community graphs; (3) **Enzymes**: 587 tertiary protein structures of enzymes from the BRENDA database (Schomburg et al., 2004); and (4) **Grid**: 100 standard 2D grid graphs (You et al., 2018).

For the general graph datasets, the three graph features used to compute the MMD distances are: *degrees* (Deg.), representing the degree distributions and local connectivity of the graphs; the *clustering coefficients* (Clus.); and *orbit counts* (Orbit), representing the distributions of the 4-orbit nodes of a graph. For the molecular datasets, RDKit was used to convert the datasets into kekulized files, remove hydrogen atoms, and replace aromatic bonds with double bonds. The generated molecules were evaluated by: the Fréchet ChemNet Distance (FCD) (Preuer et al., 2018), which calculates the distance between the generated molecules and those in the training set by using the output acquired from the penultimate activation layer of ChemNet; the neighborhood subgraph pairwise distance kernel (NSPDK) MMD (Costa & De Grave, 2010), which calculates the MMD between the generated molecules and the test set molecules, simultaneously evaluating the node and edge features; *validity w/o correction*, the validity scores of molecules without valence correction or edge resampling; *uniqueness*, the uniqueness scores of the generated molecules; and *novelty*, the proportion of generated molecules that do not appear in the training set.

B GLMM Data Flow

As illustrated in Figure 1(a), the input to the encoder consists of the node features $\mathbf{X} \in \mathbb{R}^{N \times d'}$ and edge features $\mathbf{E} \in \mathbb{R}^{N \times N \times d''}$ from the graph modality. $\mathbf{X}$ and $\mathbf{E}$ are processed separately through the node encoder (atom encoder) and the edge encoder (edge encoder), followed by a fusion encoder module to obtain the fused features. These fused features then pass through multiple transformer encoder layers to obtain the latent space modality representation $\mathbf{Z} \in \mathbb{R}^{N \times d'''}$. The decoder takes $\mathbf{Z}$ as input to the transformer decoder layer, and the node decoder (atom decoder) and edge decoder (edge decoder) finally map it back to the node features $\mathbf{X}$ and edge features $\mathbf{E}$ of the graph modality.

C Qualitative Results and Additional Ablation

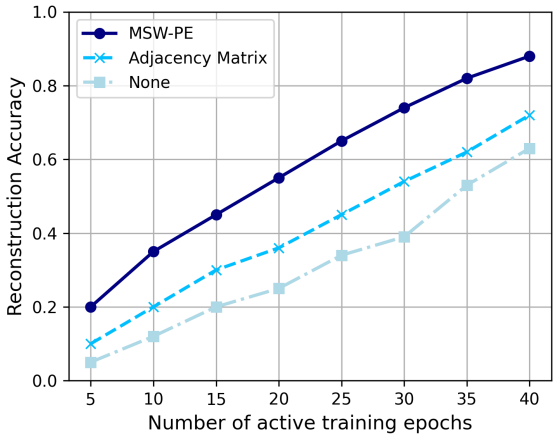


Figure 4: Comparison of the impact of different position encoding strategies (MSW-PE, adjacency matrix-based encoding, and no position encoding) on the reconstruction accuracy of the GLMM module across varying training epochs.

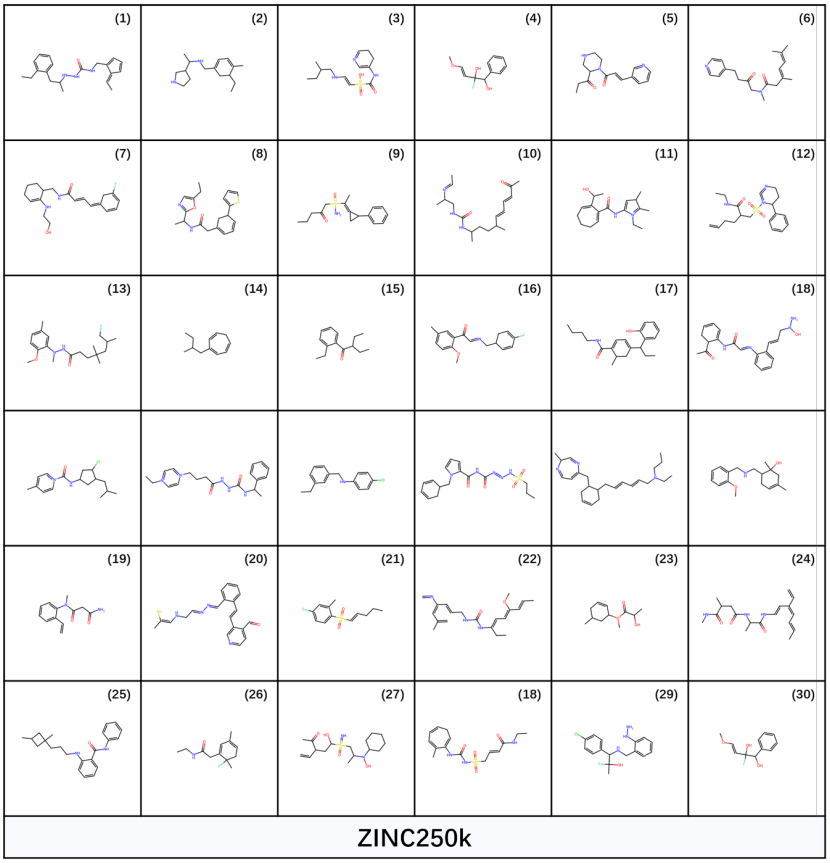


Figure 5: Visualization of generated samples on ZINC250k from GD-Mamba.

Table 3: Ablation Experiment for G-Mamba and S-Mamba Step Distributions

G-Mamba Steps	S-Mamba Steps	Validity w/o correction (%)
0	100	92.04%
10	90	94.43%
20	80	96.85%
30	70	97.94%
40	60	94.32%
50	50	91.42%
60	40	85.79%