SCORE-BASED GRAPH GENERATIVE MODELING WITH SELF-GUIDED LATENT DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph generation is a fundamental task in machine learning, and it is critical for numerous real-world applications, biomedical discovery and social science. Existing diffusion-based graph generation methods have two limitations: (i) they conduct diffusion process directly in complex graph space (i.e., node feature, adjacency matrix, or both), resulting in hard optimization with network evaluations; (ii) they usually neglect to sufficiently cover the whole distribution of target unlabeled graph set and thus fail to make semantic controllable generation. In this paper, we first propose a unified latent-based graph generative framework, Score-Based Graph Generative Model (SGGM), powered by Self-Guided Latent Diffusion (SLD) to address both limitations. Specifically, we pretrain a variational graph autoencoder to map raw graph of high-dimensional discrete space to low-dimensional topology-injected latent space, and apply score-based generative model there, yielding a smoother, faster and more expressive graph generation procedure. To sufficiently cover the whole semantical distribution of unlabeled graph set, we propose SLD to make controllable self-guidance of the sample generation with gradients from the designed assigning function towards the hierarchical pseudo label, produced by iteratively clustering on the latent embeddings. In addition, we conduct periodic update on the pseudo label in training process to achieve mutual adaptation between self-guidance and score-based generation. Experiments show that our SGGM powered by SLD outperforms previous graph generation baselines on both generic and molecular graph datasets, demonstrating the generality and extensibility along with further theoretical proofs.

1 INTRODUCTION

Graph generative models (Zhu et al., 2022; Bonifati et al., 2020; Thompson et al., 2021; Vignac & Frossard, 2021; Tang et al., 2021; O'Bray et al., 2022) are important for many real-world graph-structured applications (Deng et al., 2020; Dwivedi et al., 2022) including molecular graph generation in drug discovery (Zhang et al., 2020; Guo et al., 2022; Luo & Ji, 2021; Maziarz et al., 2021; Fu et al., 2021; Gao et al., 2021; Ahn et al., 2021; Xu et al., 2021a; Hasanzadeh et al., 2022; Adams et al., 2022; Godwin et al., 2022; Adams et al., 2022; Liu et al., 2021c), modeling physical and social interactions (Du et al., 2021; Brandstetter et al., 2021), and completing knowledge graphs. Graph generation is a challenging problem due to the complex discrete structures of graph data (Hamilton et al., 2017; Wu et al., 2020; Zhou et al., 2020; Yang et al., 2020). Graph generative models were originally studied based on the assumed structural prior (Müller et al., 1995). Then researchers found that graph generative models can be directly learned from the observation of graph set, which motivated various generative approaches (Guo & Zhao, 2020; Shirzad et al., 2022; Yang et al., 2022), such as VGAE (Kipf & Welling, 2016; Simonovsky & Komodakis, 2018), GraphRNN (You et al., 2018), GraphGAN (Wang et al., 2019; Yang et al., 2019), GraphEBM (Liu et al., 2021b), Graph Normalizing Flows (Liu et al., 2019).

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020b) is a class of generative models that have demonstrated impressive results on extensive tasks (e.g., computer vision, natural language processing, and life science) with dense theoretical founding. They treat generation tasks as a noising-denoising or destroying-restoring procedure with corresponding forward and reverse processes. Recent graph generation approaches begin to combine diffusion models to generate realistic graphs given target graph set. Some of them directly utilize the perturbations on

discrete graph space for general purpose (Niu et al., 2020a; Song et al., 2022; Gnaneshwar et al., 2022) while many others design specific diffusion processes aiming for desired properties, mostly proposed in life science (Anand & Achim, 2022; Trippe et al., 2022; Jumper et al., 2021; Luo et al., 2021a). Some choose to incorporate the geometrical properties of molecular graph (Hoogeboom et al., 2022; Xu et al., 2021b) into the diffusion process. Another line of works replace the diffusion object with torsion angles (Jing et al., 2022) or atomic coordinates (Shi et al., 2021).

Despite these diffusion-based graph generative models have achieved great performance in certain area, we argue that there are still two limitations in existing works: (i) they directly perturb nodes (Xu et al., 2021b; Gnaneshwar et al., 2022), adjacency matrix (Niu et al., 2020a), or both (Jo et al., 2022) in discrete graph space of diffusion process, and thereby can not sufficiently capture the semantical information in target graph set, leading to hard optimization with thousands of network evaluations; and (ii) their diffusion processes neglect to cover the whole distribution of graph set and thus are uncontrollable with limited expressiveness. To address these limitations, we first propose a unified latent-based score-based graph generative framework, namely Score-Based Graph Generative Model (SGGM), to overcome limitation (i). Further, we devise a new controllable diffusion mechanism Self-Guided Latent Diffusion (SLD) to overcome limitation (ii).

Instead of directly operating on nodes or edges in complex discrete graph space, **Score-Based Graph Generative Model** (SGGM) is the first to move the graph diffusion process from high-dimensional discrete graph space to low-dimensional topology-injected latent space by pretraining a VGAE with Normal prior, and then applies score-based models in this latent space. This procedure enables a smoother and faster diffusion process since SGGM only needs to optimize the score-based models in a smaller and more expressive latent space, and learns a residual distribution of latent variables with respect to the Normal prior. To guarantee an informative mapping between the latent and graph space, SGGM carefully designs its decoder with global and local matching constraints. Controllable diffusion methods (Dhariwal & Nichol, 2021; Nichol et al., 2022; Ho & Salimans, 2021) usually inject informative semantical guidance (e.g., class label) to guide the diffusion process. However, they can not be applied when labeled data is unavailable. Although customized molecule-to-conformation diffusion methods (Xu et al., 2021b; Jing et al., 2022; Hoogeboom et al., 2022) can generate conformations with desired properties, they only condition on instance-level molecular graph structure and thereby fail to cover the whole semantical distribution of (graph) data set. And their extensibility is also limited due to the specific design. The proposed **Self-Guided Latent Diffusion** (SLD) can tackle these problems. SLD first induces the hierarchical pseudo label for self-guidance through clustering the latent embeddings. Then it guides the latent generation towards the pseudo label with gradients from the designed assigning function, and iteratively injects semantical guidance in the reverse diffusion process. Notably, SLD is extensible to model many target (graph) set, including synthetic graphs, citation and social networks. To achieve mutual adaptation between self-guidance and score-based generation, we propose to conduct periodic update on the pseudo-label set in training process. Based on topology-injected latent, SGGM is unified with SLD, and we further theoretically and empirically prove the effectiveness of the proposed algorithm.

Here, we summarize our main technical contributions as follows:

- We first propose a unified latent-based graph generative framework, Score-Based Graph Generative Model (SGGM), to move the diffusion process from high-dimensional discrete graph space to low-dimensional latent space, enabling smooth, fast and expressive generation.
- We first propose a new self-guided mechanism, called Self-Guided Latent Diffusion (SLD), to enable a controllable and hierarchical graph generation procedure. It can effectively covers the whole semantical distribution of the unlabeled graph set with the designed pseudo-label assigning function.
- We theoretically prove the effectiveness of our SGGM with SLD, which also significantly outperforms previous diffusion-based and non-diffusional baselines on both generic and molecular graph generation datasets.

2 Related Work

Diffusion Models Diffusion models are new and promising deep generative models (De Bortoli et al., 2021; Dockhorn et al., 2021; Karras et al., 2022; Watson et al., 2022; Nichol & Dhariwal,

2021; Huang et al., 2021; Chen et al., 2021; Austin et al., 2021; Gu et al., 2022; Dhariwal & Nichol, 2021). The essential idea of diffusion probabilistic model (Sohl-Dickstein et al., 2015) (usually referred to as diffusion model) is to use a prefixed forward and a learnable reverse diffusion process to destroy and recover the data structure, respectively. In Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020), the forward and reverse processes are Markov chains with transition kernels in the same functional form, and it trains the reverse process by maximizing a lower bound of the log-likelihood. Some works focus on optimize the forward and reverse chains to improve model performance (Ho et al., 2020; Kingma et al., 2021; Bao et al., 2021; Song et al., 2020a). Further, Score-Based Generative Models (SGMs) study the diffusion models in continuous-time setting using stochastic differential equations (SDE) (Song et al., 2020b; 2021; Liu et al., 2021a; De Bortoli et al., 2021; Vahdat et al., 2021). SDE is utilized to transform data distribution to a known prior distribution by smoothly injecting noise, and a corresponding reverse-time SDE to reverse the transition by slowly removing the noise. In short, diffusion models experience a development process of DPM→DDPM→SGM. With respect to the utilization of diffusion models in graph generation, there are more practical problems to be addressed compared with that in computer vision tasks. A critical problem is that existing diffusion models directly the input of discrete graph space in diffusion process, and thus can not sufficiently capture the semantical information in target graph set, leading to hard optimization with thousands of network evaluations. Hence in our proposed framework SGGM, we leverage the variational graph autoencoder (Kipf & Welling, 2016; Simonovsky & Komodakis, 2018) to move the graph diffusion process from the high-dimensional discrete graph to the topology-injected latent space, and conduct a smoother and faster diffusion process there.

Graph Generative Models Graph generative models were originally proposed to generate diverse graphs based on the structural prior of target graph set (Müller et al., 1995). Modern graph generative models adopt some general generative models (Goodfellow et al., 2014; Creswell et al., 2018; Gui et al., 2021; Li et al., 2018; Vahdat & Kautz, 2020) to directly learn from the observed graph set. GraphRNN (You et al., 2018), GraphGAN (Wang et al., 2019), GraphEBM (Liu et al., 2021b) utilize Autoregressive model, Generative Adversarial Nets and Enery-Based Model respectively in graph generation tasks. However, these methods can not learn meaningful representation from observed graphs to manipulate the properties of generated graphs. In contrast, Graph Normalizing Flows (GNF) (Liu et al., 2019) and Variational Graph Autoencoder (VGAE) (Kipf & Welling, 2016) both adopt an encoder-decoder framework. GNF applies reversible message passing mechanism to encode and decode graphs, but the expressiveness of GNF is limited by the assumption of reversibility. VGAE utilizes a graph convolutional network as encoder to learn representations that encode node-level information as well as graph-level topological information, and a simple inner product as decoder. These approaches fail to make controllable generation especially with unlabeled graph set. In this paper, we first unify an enhanced VGAE architecture with diffusion models to accomplish the first latent-based graph generation. Further, we propose SLD, a self-guided latent diffusion mechanism to enable our SGGM to have a controllable graph generation procedure, and sufficiently cover the whole semantical distribution of target (unlabeled) graph set.

3 METHODOLOGY

Notations and Problem Definition. A graph with N nodes is defined as $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{A}, \mathcal{E})$, where $\boldsymbol{A} \in \{0,1\}^{N\times N}$ is the adjacent matrix (i.e. $\boldsymbol{A}_{i,j} = 1$ if there is a connection between node i and j). $\boldsymbol{X} \in \mathbb{R}^{N\times d_n}$ denotes the nodes features, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$ and $\boldsymbol{x}_i$ is the feature of node i. $\mathcal{E} \in \mathbb{R}^{N\times N\times d_e}$ denotes features of edges (i.e. $\mathcal{E}_{i,j,:}$ is the feature of edge between node i and j). For variational graph autoencoder, we use θ_{enc} and θ_{dec} to denote the parameters of encoder f_{enc} and decoder f_{dec} respectively. The latent variables produced by the encoder is $\boldsymbol{Z} \in \mathbb{R}^{N\times F}$ with $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N]$. For diffusion models, we use $\boldsymbol{Z}_0$ to denote the input latent variable, and $\boldsymbol{Z}_t$ for $t \in [0, T]$ to denote the variable in the diffusion models at time t. The aim of graph generation task is to learn a graph generative model that captures the distribution of target graph set in training process, and thus the learned model can generate realistic graph $\mathcal{G}$ in evaluation.

Now we presents SGGM, a first unified latent-based graph generative framework to address graph generation tasks, which is illustrated in subsection 3.1. In subsection 3.2, we introduce a novel self-guided diffusion mechanism SLD to help SGGM modeling the whole distribution of target graph

set. And we provide the optimization detail in training process and summarize the entire algorithm. The overall framework SGGM with SLD is summarized in Fig.1.

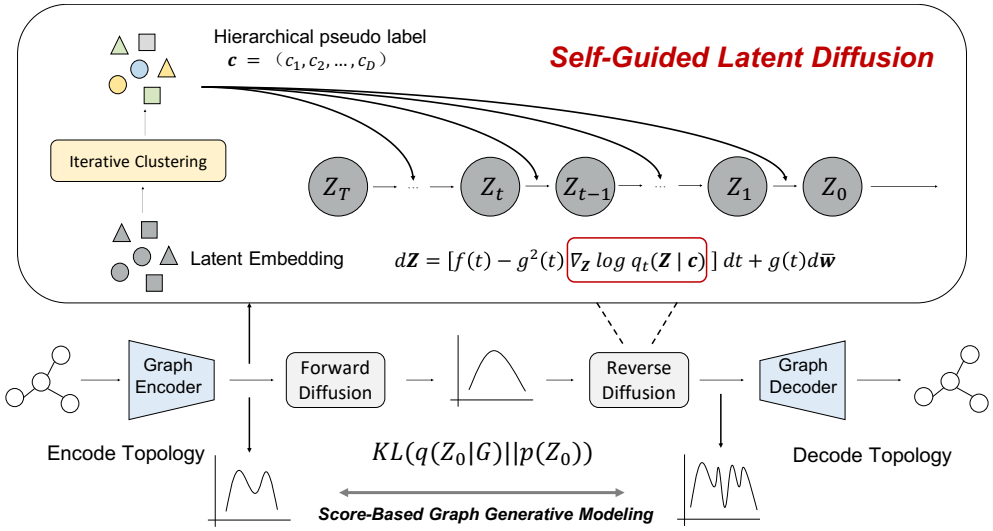

Figure 1: Schematic illustration of the proposed SGGM with SLD.

3.1 Score-Based Graph Generative Model

Motivations. Based on the generative diffusion models, diffusion-based graph generative models follow a nosing-denoising or destroying-restoring procedure in a graph space. Existing approaches choose to add noises on the input nodes, perturb input adjacency matrix or corrupt both of them, and then learn to recover the original inputs by modeling complex dependency between nodes and edges. Such forward-backward training process is hard to optimize since it is conducted on a high-dimensional complex discrete space, and thus requires costly sampling with thousands of network evaluations. Besides, most of diffusion-based graph generative models propose a specific architecture for certain domain (e.g, molecular conformation generation), which limits the extensibility. Therefore, we need to design a unified diffusion-based framework that is not only easy to optimize, but also extensible to more graph generation tasks.

From Graph Space to Topology-Injected Latent Space. We here introduce our unified diffusion-based graph generative framework in detail. The overall pipeline experiences a summarized procedure of graph-to-latent, latent-to-latent, and latent-to-graph. This framework first combines VGAE with score-based generative models to solve various graph generation tasks from a unified perspective. To relax the diffusion-based graph generation process, SGGM pretrains a VGAE to map raw graph $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{A}, \mathcal{E})$ of high-dimensional discrete space to low-dimensional topology-injected latent $\mathbf{H}$. The encoding process can be summarized as follows:

$$\boldsymbol{H} = f_{enc}(\boldsymbol{X}, \mathcal{E}; \theta_{enc}), \tag{1}$$

where f_{enc} can be variants of graph neural networks, e.g., GCN, GAT, and GIN. θ_{enc} denotes all the parameters in the encoding process. For each node latent $\boldsymbol{h}_i$ in $\boldsymbol{H}$, it is obtained by layer-wise message passing. For $l \in 0, 1, ...L$, we perform the following transformation at layer l ($\boldsymbol{h}_0 = \boldsymbol{x}_0$):

$$\boldsymbol{m}_i^l = \sum_{j \in \mathcal{N}(i)} M_l(\boldsymbol{h}_i^{l-1}, \boldsymbol{h}_j^{l-1}, \mathcal{E}_{i,j}; \theta_{enc}^l), \quad \boldsymbol{h}_i^l = U_l(\boldsymbol{h}_i^{l-1}, \boldsymbol{m}_i^l), \quad \forall i, \tag{2}$$

where $\mathcal{N}(i)$ denotes the nodes connected to node i, M_l and U_l are message passing function and update function, respectively. After L layers, we use the final representations to produce latent variable $\boldsymbol{Z}_0$ with reparameterization trick (Tomczak & Welling, 2018):

$$\mu(\boldsymbol{H}) = \boldsymbol{W}_1 \boldsymbol{H}, \quad \log \sigma(\boldsymbol{H}) = \boldsymbol{W}_2 \boldsymbol{H}, \tag{3}$$

$$\boldsymbol{Z}_0 = \mathcal{N}(\boldsymbol{Z}_0 | \mu(\boldsymbol{H}),\ \sigma^2(\boldsymbol{H})), \tag{4}$$

where $W_1, W_2 \in \mathbb{R}^{N\times N}$. In this way, we make this mapping process differentiable and enables an end-to-end training architecture.

Diffusion in Topology-Injected Latent Space. After acquiring topology-injected latent space, a score-based diffusion process is conducted to model the residual distribution of the latent variable with respect to the Normal prior. By optimizing in a smaller space, the generative diffusion process can be smoother and faster. Conversely, diffusion models can also help VGAE to parameterize the prior over the latent variables, boosting the performance and expressiveness of graph generation. Formally, considering the mismatch between latent distribution $p_\theta(\boldsymbol{Z}_0)$ and the Normal prior, the forward diffusion process is formulated as the following:

$$d\boldsymbol{Z} = f(t)\boldsymbol{Z}dt + g(t)d\mathbf{w}, \quad \boldsymbol{Z}_0 \sim p_\theta(\boldsymbol{Z}_0) \tag{5}$$

with prefixed drift and diffusion coefficient $f(t)$ and $g(t)$, resulting in a tractable prior $\mathbf{Z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Therefore, it effectively reduces the mismatch between the marginal distribution of $\boldsymbol{Z}_0$ and Normal prior. And the backward diffusion process progressively transforms $\boldsymbol{Z}_T$ to $\boldsymbol{Z}_0$ using the reverse SDE:

$$d\hat{\boldsymbol{Z}} = (f(t)\hat{\boldsymbol{Z}} - g(t)^2\nabla_{\hat{\boldsymbol{Z}}}\log q_t(\hat{\boldsymbol{Z}}))dt + g(t)d\bar{\mathbf{w}}. \tag{6}$$

Compared with previous methods that conduct diffusion in raw discrete graph space, SGGM can lead to more expressive and meaningful graph generation conditioned on topological information. The diffusion process and the encoder in SGGM are trained together by aligning the approximate posterior $q_{\theta_{enc}}(\boldsymbol{Z}_0|\mathcal{G})$ to the diffusion prior $p_\theta(\boldsymbol{Z}_0)$ with Kl divergence:

$$\begin{aligned}\mathcal{L}_{diff} &= D_{\mathrm{KL}}(q_{\theta_{enc}}(\boldsymbol{Z}_0|\mathcal{G})||p_\theta(\boldsymbol{Z}_0)) &(7)\\ &= \mathbb{E}_{q_{\theta_{enc}}(\boldsymbol{Z}_0|\mathcal{G})}\log q_{\theta_{enc}}(\boldsymbol{Z}_0|\mathcal{G}) - \mathbb{E}_{q_{\theta_{enc}}(\boldsymbol{Z}_0|\mathcal{G})}\log p_\theta(\boldsymbol{Z}_0) &(8)\end{aligned}$$

And the cross entropy term in Eq.(8) can be further simplified as:

$$\mathbb{E}_{t\sim\mathcal{U}(0,T)}\left[\frac{g(t)^2}{2}\mathbb{E}_{q(\boldsymbol{Z}_0,\boldsymbol{Z}_t|\mathcal{G})}[||\nabla\log q(\boldsymbol{Z}_t|\boldsymbol{Z}_0) - \mathbf{s}_\theta(\boldsymbol{Z}_t,t)||_2^2]\right] + C \tag{9}$$

with $q(\boldsymbol{Z}_0,\boldsymbol{Z}_t|\mathcal{G}) = q_{\theta_{enc}}(\boldsymbol{Z}_0|\mathcal{G})q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)$ and $q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)$ is gaussian distributed according to Equation (5). The motivation of this objective is the intractability of the classical score-matching objectives for diffusion in latent space, which is theoretically proved in Appendix.A.2.

From Topology-Injected Latent Space to Graph Space. To recover the original graph, we apply both global and local matching constraints on our graph decoder, which is different from original VGAE. It is critical to carefully design the graph decoder since its influence will pass back to the score-based diffusion process. To this end, we not only force the latent embedding to decode the direct links (adjacency matrix) by minimizing global reconstruction loss, but also recover its node feature $\boldsymbol{X}$ and degree of nodes $\boldsymbol{X}_d$ for local structural reconstruction. Given the latent variable $\boldsymbol{Z}_0 = [\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_N]$ as the input of the decoder f_{dec} where θ_{dec} denotes all the parameters in the decoding process, and we can formulate the decoding process as follows:

$$\hat{\boldsymbol{A}}, \hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}_d = f_{dec}(\boldsymbol{Z}_0, \boldsymbol{X}, \boldsymbol{A}; \theta_{dec}), \tag{10}$$

$$\hat{\boldsymbol{A}} = \sigma(\boldsymbol{Z}_0^T\boldsymbol{Z}_0), \quad \hat{\boldsymbol{X}} = \hat{\boldsymbol{A}}\mathrm{ReLU}(\hat{\boldsymbol{A}}\boldsymbol{Z}_0\boldsymbol{W}_3)\boldsymbol{W}_4, \quad \hat{\boldsymbol{X}}_d = \mathrm{ReLU}(\hat{\boldsymbol{X}}\boldsymbol{W}_5), \tag{11}$$

where $\boldsymbol{W}_3$, $\boldsymbol{W}_4$, and $\boldsymbol{W}_5$ are all linear transformation matrices. $\hat{\boldsymbol{X}}$, $\hat{\boldsymbol{A}}$, and $\hat{\boldsymbol{X}}_d$ are the reconstructed node features, reconstructed topological connections, and predicted degree of nodes, respectively. Based on outputs, we make the following regularization:

$$\mathcal{L}_{rec} = \mathbb{E}_{q_{\theta_{enc}}(\boldsymbol{Z}|\mathcal{G})}\left[-\log\prod_{i=1}^{N}\prod_{j=1}^{N}p_{\theta_{dec}}(\hat{\boldsymbol{A}}_{i,j}|\boldsymbol{z}_i,\boldsymbol{z}_j) + \alpha||\boldsymbol{X} - \hat{\boldsymbol{X}}||^2 + \beta||\boldsymbol{X}_d - \hat{\boldsymbol{X}}_d||^2\right], \tag{12}$$

where α and β are weights for balancing the terms. Particularly, incorporating degree prediction helps recovering the local structures. And such global and local matching constraints tailored for diffusion process will also make graph generation more expressive. The above is all the processes of our unified latent-based graph diffusion framework SGGM. Next, we will introduce a new self-guided mechanism, which is unified with SGGM based on the topology-injected latent.

3.2 SELF-GUIDED LATENT DIFFUSION

Motivations. Controllable diffusion-based generation (Ramesh et al., 2022; Dhariwal & Nichol, 2021; Ho & Salimans, 2021; Nichol et al., 2022) is to introduce informative guidance in reverse process, which has shown promising results in vision tasks. Nevertheless, they usually condition on the class label to make controllable generation, and thereby fail to effectively model the whole semantical distribution of the unlabeled graph set. Despite some diffusion-based conditional graph generation methods have been proposed (Xu et al., 2021b; Jing et al., 2022; Hoogeboom et al., 2022), they mostly aims for molecule-to-conformation generation conditioned on instance-level graph structure and thereby also fail in modeling the whole semantical distribution. Besides, they are limited in extensibility due to the specific design. Hence, we propose a new self-guidance mechanism SLD to cover the whole distribution of unlabeled target graph set, and thus enable controllable and hierarchical graph generation, further improving expressiveness of the proposed SGGM.

Self-Guided Latent Diffusion. Specifically, we conduct iterative K-means clustering on all the latent embeddings of the graph set, and consequently assign a hierarchical pseudo class label vector $\boldsymbol{c}$ for each graph latent variable $\boldsymbol{Z}_0$:

$$\boldsymbol{c} = (c_1, c_2, \cdots, c_D), \quad 1 \leq c_i \leq \mathcal{P}_i, \quad \forall i \leq D, \tag{13}$$

where c_i and $\mathcal{P}_i$ denote the label index and the total number of classes in i-th hierarchy, respectively. After categorizing latent variables in SGGM with pseudo label, we bring this informative guidance into the diffusion process. For better understanding the mechanism of our proposed self-guided latent diffusion, we theoretically provide a detailed explanation, please refer to Appendix.A.3. Such self-guidance can improve the sample quality and diversity by guiding the generation process for $\boldsymbol{Z}_0$ towards the region of class $\boldsymbol{c}$. At each generation step, the score function of class conditional diffusion model $\mathbf{s}_\theta(\boldsymbol{Z}_t, t, \boldsymbol{c})$ is modified to incorporate the gradient information of $\log q_t(\boldsymbol{c}|\boldsymbol{Z})$:

$$\hat{\mathbf{s}}_\theta(\boldsymbol{Z}_t, t, \boldsymbol{c}) = \mathbf{s}_\theta(\boldsymbol{Z}_t, t, \boldsymbol{c}) - w\nabla_{\boldsymbol{Z}} \log q_t(\boldsymbol{c}|\boldsymbol{Z}_t), \tag{14}$$

and use $\hat{\mathbf{s}}_\theta(\boldsymbol{Z}_t, t, \boldsymbol{c})$ in generation instead, where w controls the magnitude of the guidance. An alternative way is to directly estimate $\nabla_{\boldsymbol{Z}} \log q(\boldsymbol{c}|\boldsymbol{Z}_t)$, but it need to first train a class-conditional diffusion model, and then train a classifier to predict c for each $(\boldsymbol{Z}_t, t)$ to calculate the gradient information. Such sophisticated procedure will enlarge the network evaluation steps. Hence, we adopt a classifier-free approach instead and use Bayes rule to calculate the self-guidance, by jointly learning an unconditional and a pseudo-label-conditional score function. With the conditional score $\mathbf{s}_\theta(\boldsymbol{Z}_t, t, \boldsymbol{c}) \approx \nabla \log q(\boldsymbol{Z}_t|\boldsymbol{c})$ and unconditional score $\mathbf{s}_\theta(\boldsymbol{Z}_t, t) \approx \nabla \log q(\boldsymbol{Z}_t)$, the guidance can be calculated as Eq.(15) and the resulting reverse SDE in Eq.(6) can be rewritten as Eq.(16):

$$\nabla_{\boldsymbol{Z}} \log q_t(\boldsymbol{c}|\boldsymbol{Z}) = \nabla_{\boldsymbol{Z}} \log q_t(\boldsymbol{Z}|\boldsymbol{c}) - \nabla_{\boldsymbol{Z}} \log q_t(\boldsymbol{Z}) \approx \mathbf{s}_\theta(\boldsymbol{Z}_t, t, \boldsymbol{c}) - \mathbf{s}_\theta(\boldsymbol{Z}_t, t) \tag{15}$$

$$d\hat{\boldsymbol{Z}} = (f(t)\hat{\boldsymbol{Z}} - g(t)^2[(1-w)(\mathbf{s}_\theta(\hat{\boldsymbol{Z}}_t, t, \boldsymbol{c}) + w\mathbf{s}_\theta(\hat{\boldsymbol{Z}}_t, t)]dt + g(t)d\bar{\mathbf{w}}. \tag{16}$$

Then we use a single neural network to parameterize both models, where for the unconditional model we can simply input a zero vector $\mathbf{0}$ for the pseudo label $\boldsymbol{c}$ when estimating the score, i.e., $\mathbf{s}_\theta(\boldsymbol{Z}_t, t) = \mathbf{s}_\theta(\boldsymbol{Z}_t, t, \mathbf{0})$. In this way, SLD iteratively injects global semantical guidance into the reverse diffusion process and further improves SGGM with controllable generation. Besides, SLD can further cover the whole semantical distribution in sampling generation procedure with sufficient expressiveness. Notably, the proposed SLD is also extensible to any target graph set modeling, which is particularly superior to existing conditional molecule-to-conformation diffusion methods (Xu et al., 2021b; Jing et al., 2022; Huang et al., 2022).

Periodic Update on Pseudo-Label Set. Nevertheless, there exists a disalignment between self-guidance and score-based generation procedure. Here, we explain why this disalignment comes up. The produced global pseudo label set $\{\{c_{i,d}\}_{i=1}^{\mathcal{P}_i}\}_{d=1}^{D}$ in SLD is induced by iterative K-means clustering on embedding vectors in latent space of SGGM, which is determined value and thus invariant. D is the total number of hierarchies. In contrast, as the training process goes on, the latent space out of the encoder in SGGM will progressively shifts. Therefore, the invariant pseudo label and shifted latent space will consequently result in the disalignment, and further deteriorate the performance of SGGM. To decrease the alignment, we propose to apply the periodic update on the pseudo-label set by conducting iterative K-means clustering based on updated encoder every certain training steps. In the evaluation procedure, we use the pseudo-label set in the last training step to

make self-guided graph generation. And we provide the overall optimization objective for SGGM with SLD and summarize the loss terms in Eq.(8) and Eq.(12) as the following:

$$\mathcal{L}(\mathcal{G};\theta,\theta_{enc},\theta_{dec}) = \mathcal{L}_{diff} + \mathcal{L}_{rec}. \tag{17}$$

Overall objective is an upper bound for the negative log-likelihood of generated sample, and we further provide an efficient alternative formulation of the objective, as demonstrated in Appendix.A.1. To clear the training pipeline, we summarize the entire procedure in Algorithm 1.

Algorithm 1 Algorithm of SGGM powered by SLD

Input: Target graph set $\mathbf{G} = \{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_K\}$, score-based models $\boldsymbol{s}_\theta$, variational graph autoencoder (f_{enc}, f_{dec}), training steps S, update step size U, sampling steps M, sampling step size δt.
Output: Trained models $\boldsymbol{s}_\theta$, f_{enc}, f_{dec}.

Initialize the f_{enc} and f_{dec} by pretraining on $\mathbf{G}$.
Hierarchically cluster to obtain global pseudo label set $\{\{c_{i,d}\}_{i=1}^{\mathcal{P}_i}\}_{d=1}^{D}$.
for $s = 1$ **to** S **do**
  if $s = U$ **then**
    Update global pseudo label set $\{\{c_{i,d}\}_{i=1}^{\mathcal{P}_i}\}_{d=1}^{D}$.
  end if
  Sample a mini-batch graph $\{\mathcal{G}_i\}_{i=1}^{Q} \in \mathbf{G}$.
  for $i = 1$ **to** Q **do**
    Map graph $\mathcal{G}_i$ to latent $\boldsymbol{Z}$ with Eq.(1), (3), and (4).
    for $j = 1$ **to** M **do**
      Use δt and $\boldsymbol{Z}$ to make forward diffusion on latent with Eq.(5).
    end for
    for $j = 1$ **to** M **do**
      Use δt, $\{\{c_{i,d}\}_{i=1}^{\mathcal{P}_i}\}_{d=1}^{D}$, $\boldsymbol{s}_\theta$ and $\boldsymbol{Z}$ to make self-guided reverse diffusion with Eq.(16).
    end for
    Map latent $\boldsymbol{Z}$ to graph by reconstructing $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{H}})$ with Eq.(10) and (11).
  end for
  Update parameters of $\boldsymbol{s}_\theta$, f_{enc}, f_{dec} according to Eq.(17).
end for
Return: $\boldsymbol{s}_\theta$, f_{enc}, f_{dec}

4 Experiments and Analysis

Graph Dataset Details. In this paper, we adopt six widely-used graph datasets to evaluate our method, following previous works (You et al., 2018; Jo et al., 2022; Xu et al., 2021b; Du et al., 2021). These graph datasets includes various domains, such as synthetic graphs, social networks, and molecular graphs: `Ego-small` has 200 small ego graphs that collected from a large citeseer graph dataset (Sen et al., 2008); `Community-small` collects 100 randomly generated social community graphs; `Enzymes` has 587 protein graphs which represent the protein tertiary structures of the enzymes from the BRENDA database (Schomburg et al., 2004); `Grid` has 100 standard 2D grid graphs; `QM9` (Ramakrishnan et al., 2014) has 133,885 molecular graphs with 4 node types; and `ZINC250k` (Irwin et al., 2012) has 249,455 molecular graphs with 9 node types.

Evaluation Metrics. We use two main categories of metrics for model evaluation, following You et al. (2018); Du et al. (2021) for fair comparisons. One is the statistics-based evaluation metrics for generic graph generation, another is the quality-based evaluation metrics. Four **Statistics-based evaluation metrics** are introduced as follows: *node degree distribution* denotes the empirical node degree distribution of a graph, which represents its local connectivity patterns; *clustering coefficient distribution* denotes the empirical clustering coefficient distribution of a graph; and *orbit count distribution* denotes the distribution of the counts of node 4-orbits of a graph. Four **Quality-based evaluation metrics** are introduced as follows: *Fréchet ChemNet Distance (FCD)* (Preuer et al., 2018) is to evaluate the distance between the training and generated graph sets using the activations of the penultimate layer of the ChemNet; *Neighborhood subgraph pairwise distance kernel (NSPDK) MMD* (Costa & De Grave, 2010) calculates the MMD between the generated and test molecules

which considers both the node and edge features for evaluation. These two metrics are utilized to effectively measure how close the generated molecules rely on the distribution and how exactly the model capture the distribution of target graph sets. Besides, validity w/o correction is to calculate the fraction of valid molecules without correcting valency or resampling edge. *Time* is to measure the spending time for generating 10,000 molecules in the form of RDKit molecules.

Baseline Methods We compare our proposed method against following **general graph generative models**. DeepGMG (Li et al., 2018) and GraphRNN (You et al., 2018) adopt RNN-based architectures while GraphAF (Shi et al., 2020) and GraphDF (Luo et al., 2021b) apply flow-based architectures. Above models are all autoregressive, generating graph step by step. GraphVAE (Simonovsky & Komodakis, 2018), GraphEBM Liu et al. (2021b), GDSS (Jo et al., 2022) and EDP-GNN (Niu et al., 2020a) utilize VAE, EBM, and score-based models respectively while GNF (Liu et al., 2019) and MoFlow (Zang & Wang, 2020) deploy flow-based model. Above models are all one-shot, generating the entire graph in one step. From a perspective of architecture, our SGGM first incorporates VGAE with score-based model to address graph generation tasks and is also an efficient one-shot generative model. More implementation details and hyperparameter settings are in Appendix.B.1.

Table 1: Comparison results of generic graph generation on Ego-small, Community-small, Enzymes, and Grid datasets. We calculate the MMD distances between the test datasets and generated graphs. All reported results of previous baselines are quoted from published papers (Niu et al., 2020b; Luo et al., 2021b; Jo et al., 2022) or reproduced by published codes. Best results are highlighted in bold (the smaller the better) and underline denotes the second best. Due to the space limitation, we show the standard deviations of results in Appendix.B.2.

Methods	Ego-small				Community-small				Enzymes				Grid			
	Real, $4 \leq \|V\| \leq 18$				Synthetic, $12 \leq \|V\| \leq 20$				Real, $10 \leq \|V\| \leq 125$				Synthetic, $100 \leq \|V\| \leq 400$			
	Deg.	Clus.	Orbit	Avg.	Deg.	Clus.	Orbit	Avg.	Deg.	Clus.	Orbit	Avg.	Deg.	Clus.	Orbit	Avg.
DeepGMG	0.040	0.100	0.020	0.053	0.220	0.950	0.400	0.523	-	-	-	-	-	-	-	-
GraphRNN	0.090	0.220	0.003	0.104	0.080	0.120	0.040	0.080	**0.017**	0.062	0.046	0.042	**0.064**	0.043	**0.021**	**0.043**
GraphAF	0.03	0.11	**0.001**	0.047	0.18	0.20	0.02	0.133	1.669	1.283	0.266	1.073	-	-	-	-
GraphDF	0.04	0.13	0.01	0.060	0.06	0.12	0.03	0.070	1.503	1.061	0.202	0.922	-	-	-	-
GraphVAE	0.130	0.170	0.050	0.117	0.350	0.980	0.540	0.623	1.369	0.629	0.191	0.730	1.619	**0.0**	0.919	0.846
GNF	0.030	0.100	**0.001**	0.044	0.200	0.200	0.110	0.170	-	-	-	-	-	-	-	-
EDP-GNN	0.052	0.093	0.007	0.051	0.053	0.144	0.026	0.074	0.023	0.268	0.082	0.124	0.455	0.238	0.328	0.340
GDSS	0.021	0.024	0.007	0.017	0.045	0.086	**0.007**	0.046	0.026	**0.061**	0.009	0.032	0.111	0.005	0.070	0.062
SGGM	0.025	0.028	0.009	0.021	0.041	0.079	0.010	0.043	0.030	0.073	0.013	0.039	0.114	**0.0**	0.065	0.060
SGGM+SLD	**0.014**	**0.019**	0.007	**0.013**	**0.035**	**0.071**	**0.006**	**0.037**	0.022	**0.062**	**0.007**	**0.030**	0.103	**0.0**	0.053	0.052

Generic Graph Generation. We show the comparison results about generic graph generation in Tab.1, and we conclude that the proposed SGGM significantly outperform most of the previous baselines including both diffusion-based and non-diffusional graph generative models. Besides, equipped with SLD, our SGGM can achieve higher performance, demonstrating the strength of the proposed self-guidance mechanism. Notably, compared with previous state-of-the-art diffusion-based models EDP-GNN and GDSS, our SGGM with SLD has superior performances in both small and large graph generation tasks. This phenomenon shows that our model is able to sufficiently cover the whole distribution of target graph set, including those containing complex global or fine-grained graph structures. Visualization results of small, middle, and large generated graphs are in Appendix.C. More implementation details are in Appendix.B.1.

Molecular Graph Generation. We further show the comparison results with previous baselines about molecular graph generation, for evaluating the extension to distribution modeling of specific graph set. Results are listed in Tab.2, we conclude that our SGGM with SLD also outperform previous general graph generative baselines. The first observation is our model achieves the best validadity when the post-hoc valency correction is disabled, showing that our latent-based framework with carefully-designed decoder helps with learning precise chemical structural information. The second observation is SGGM with SLD significantly outperforms most of the baselines in NSPDK MMD and FCD, demonstrating that the proposed SLD assists SGGM in covering the whole distribution of the target molecular graph set in both topological and chemical space. More results of validity, uniqueness, and novelty along with the standard deviations in Appendix.B.2. Notably, SGGM costs fewer time in sampling procedure than most of previous methods, especially diffusion-based methods due to the smoother and faster latent diffusion process. We also visualize some generated molecular graphs in Fig.2 and observe that the generated graphs reveal the obvious hierarchy in structures, further proving the expressiveness of our model. More implementation details are in Appendix.B.1.

Table 2: Comparison results of molecular graph generation on QM9 and ZINC250k datasets. Results are the mean value of 3 different runs and all reported results of baselines are quoted from published papers (Niu et al., 2020b; Luo et al., 2021b; Jo et al., 2022) or reproduced by published codes. Due to the space limitation, we show the results of validity, uniqueness, and novelty along with the standard deviations in Appendix.B.2

Method	QM9				ZINC250k			
	Val. w/o corr. (%)↑	NSPDK↓	FCD↓	Time (s)↓	Val. w/o corr. (%)↑	NSPDK↓	FCD↓	Time (s)↓
GraphAF	67.62	0.020	5.268	$2.52e^3$	68.47	0.044	16.289	$5.80e^3$
GraphDF	82.67	0.063	10.816	$5.35e^4$	89.03	0.176	34.202	$6.03e^4$
MoFlow	91.36	0.017	4.467	**4.60**	63.11	0.046	20.931	$\mathbf{2.45e^1}$
EDP-GNN	47.52	0.005	2.680	$4.40e^3$	82.97	0.049	16.737	$9.09e^3$
GraphEBM	8.22	0.030	6.143	$3.71e^1$	5.29	0.212	35.471	$5.46e^1$
GDSS	95.72	**0.003**	2.900	$1.14e^2$	97.01	0.019	14.656	$2.02e^3$
SGGM	95.91	0.006	2.745	$4.93e^1$	97.28	0.018	13.931	$1.01e^3$
SGGM+SLD	**97.35**	**0.004**	**2.593**	$5.43e^1$	**98.32**	**0.014**	**11.379**	$1.12e^3$

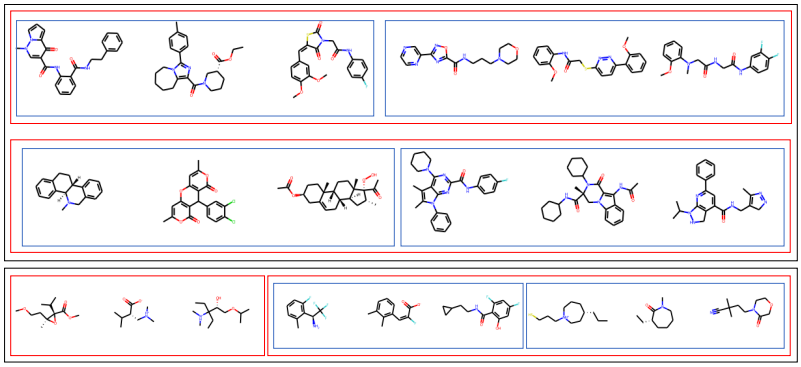

Figure 2: Visualization of generated molecular graphs on ZINC250k dataset with the proposed self-guided latent diffusion. Blue, red, black boxes denote different hierarchies.

Ablation Study and Extensibility Analysis. To clarify the contribution of different modules in our model and further show the extensibility of our model, we conduct experiments and show the results in Tab.3. In ablation part, we find that designed matching constraints and SLD can both improve the baseline (combination of VGAE and SGM without additions), and SLD contributes more. In extension part, we observe that our SGGM with SLD extended with geometrical and torsional diffusion can obtain further promotion, demonstrating the great extensibility of our model.

Table 3: Ablation study and extensibility analysis.

Datasets	Ablation			Extension	
	Baseline	+ Designed Matching	+ SLD	+ Geometry (Xu et al., 2021b)	+ Torsion (Jing et al., 2022)
ZINC250k	96.72	97.28	**98.32**	98.32 + 0.3	98.32 + 0.4
QM9	95.42	95.91	**97.35**	97.35 + 0.5	97.35 + 0.4

5 Conclusion

In this paper, we first propose a unified latent-based graph generative framework, Score-Based Graph Generative Model (SGGM), powered by Self-Guided Latent Diffusion (SLD) to address graph generation tasks. We first map raw graph of high-dimensional discrete space to low-dimensional topology-injected latent space, and apply score-based generative model there, yielding a smoother, faster and more expressive graph generation procedure. To sufficiently cover the whole distribution of graph set, we propose SLD to make controllable self-guidance of the graph generation with gradients from the designed assigning function towards the hierarchical pseudo label, produced by iteratively clustering on the latent embeddings. We theoretically prove the effectiveness of our model and we significantly outperform previous baselines on both generic and molecular graph datasets. For the future work, we will continue to promote the self-guided diffusion mechanism and further improve the expressiveness of graph generative models.

REFERENCES

Keir Adams, Lagnajit Pattanaik, and Connor W. Coley. Learning 3d representations of molecular chirality with invariance to bond rotations. In *International Conference on Learning Representations*, 2022.

Sungsoo Ahn, Binghong Chen, Tianzhe Wang, and Le Song. Spanning tree-based graph generation for molecules. In *International Conference on Learning Representations*, 2021.

Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2021.

Angela Bonifati, Irena Holubová, Arnau Prat-Pérez, and Sherif Sakr. Graph generators: State of the art and open challenges. *ACM Computing Surveys (CSUR)*, 53(2):1–30, 2020.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e (3) equivariant message passing. In *International Conference on Learning Representations*, 2021.

Tianrong Chen, Guan-Horng Liu, and Evangelos Theodorou. Likelihood training of schrödinger bridge using forward-backward sdes theory. In *International Conference on Learning Representations*, 2021.

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

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1): 53–65, 2018.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Fei Deng, Zhuo Zhi, Donghun Lee, and Sungjin Ahn. Generative scene graph networks. In *International Conference on Learning Representations*, 2020.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *International Conference on Learning Representations*, 2021.

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.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

Tianfan Fu, Wenhao Gao, Cao Xiao, Jacob Yasonik, Connor W Coley, and Jimeng Sun. Differentiable scaffolding tree for molecule optimization. In *International Conference on Learning Representations*, 2021.

Wenhao Gao, Rocío Mercado, and Connor W Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. In *International Conference on Learning Representations*, 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.

Jonathan Godwin, Michael Schaarschmidt, Alexander L Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple GNN regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696–10706, 2022.

Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Minghao Guo, Veronika Thost, Beichen Li, Payel Das, Jie Chen, and Wojciech Matusik. Data-efficient graph grammar learning for molecular generation. In *International Conference on Learning Representations*, 2022.

Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *arXiv preprint arXiv:2007.06686*, 2020.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, 2017.

Arman Hasanzadeh, Ehsan Hajiramezanali, Nick Duffield, and Xiaoning Qian. Morel: Multi-omics relational learning. In *International Conference on Learning Representations*, 2022.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34: 22863–22876, 2021.

Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant graph mechanics networks with constraints. In *International Conference on Learning Representations*, 2022.

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.

Bowen Jing, Gabriele Corso, Regina Barzilay, and Tommi S Jaakkola. Torsional diffusion for molecular conformer generation. In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10362–10383. PMLR, 2022.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *ArXiv*, abs/2206.00364, 2022.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

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

Greg Landrum et al. Rdkit: Open-source cheminformatics software, 2016. *URL http://www. rdkit. org/, https://github. com/rdkit/rdkit*, 2016.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph normalizing flows. *Advances in Neural Information Processing Systems*, 32, 2019.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021a.

Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. Graphebm: Molecular graph generation with energy-based models. In *Energy Based Models Workshop-ICLR 2021*, 2021b.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2021c.

Shitong Luo, Chence Shi, Minkai Xu, and Jian Tang. Predicting molecular conformation via dynamic graph score matching. *Advances in Neural Information Processing Systems*, 34:19784–19795, 2021a.

Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International Conference on Learning Representations*, 2021.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation. *International Conference on Machine Learning*, 2021b.

Krzysztof Maziarz, Henry Richard Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs. In *International Conference on Learning Representations*, 2021.

Berndt Müller, Joachim Reinhardt, and Michael T Strickland. *Neural networks: an introduction*. Springer Science & Business Media, 1995.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

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, 2020a.

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

Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. In *International Conference on Learning Representations*, 2022.

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.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 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 Res.*, 32(Database-Issue):431–433, 2004.

Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 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.

Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, pp. 9558–9568. PMLR, 2021.

Hamed Shirzad, Hossein Hajimirsadeghi, Amir H Abdi, and Greg Mori. Td-gen: Graph generation using tree decomposition. In *International Conference on Artificial Intelligence and Statistics*, pp. 5518–5537. PMLR, 2022.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, 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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020b.

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.

Yang Song, Qiyu Kang, Sijie Wang, Zhao Kai, and Wee Peng Tay. On the robustness of graph neural diffusion to topology perturbations. *arXiv preprint arXiv:2209.07754*, 2022.

Mingyue Tang, Pan Li, and Carl Yang. Graph auto-encoder via neighborhood wasserstein reconstruction. In *International Conference on Learning Representations*, 2021.

Rylee Thompson, Boris Knyazev, Elahe Ghalebi, Jungtaek Kim, and Graham W Taylor. On evaluation metrics for graph generative models. In *International Conference on Learning Representations*, 2021.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223. PMLR, 2018.

Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Clement Vignac and Pascal Frossard. Top-n: Equivariant set and graph generation without exchangeability. In *International Conference on Learning Representations*, 2021.

Hongwei Wang, Jialin Wang, Jia Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Wenjie Li, Xing Xie, and Minyi Guo. Learning graph representation with generative adversarial nets. *IEEE Transactions on Knowledge and Data Engineering*, 33(8):3090–3103, 2019.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. *ArXiv*, abs/2202.05830, 2022.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations*, 2021a.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2021b.

Carl Yang, Peiye Zhuang, Wenhan Shi, Alan Luu, and Pan Li. Conditional structure generation through graph variational generative adversarial nets. *Advances in Neural Information Processing Systems*, 32, 2019.

Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13390–13399, 2020.

Ling Yang, Zhilong Zhang, and Shenda Hong. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

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.

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.

Dongsu Zhang, Changwoon Choi, Jeonghwan Kim, and Young Min Kim. Learning to generate 3d shapes with generative cellular automata. In *International Conference on Learning Representations*, 2020.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Yanqiao Zhu, Yuanqi Du, Yinkai Wang, Yichen Xu, Jieyu Zhang, Qiang Liu, and Shu Wu. A survey on deep graph generation: Methods and applications. *arXiv preprint arXiv:2203.06714*, 2022.

A PROOFS

A.1 DERIVATION OF EQUIVALENT EFFICIENT OPTIMIZATION OBJECTIVE

We first demonstrate that our loss function is an upper bound of the negative log-likelihood.

$$\begin{aligned}
-\log p(\mathcal{G}) &\leq KL(p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})||q(\boldsymbol{Z}|\mathcal{G};\theta_{dec}) - \log p(\mathcal{G}) \\
&= -\mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log q(\boldsymbol{Z}|\mathcal{G};\theta_{dec}) - \log p(\mathcal{G}) + \mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log p(\boldsymbol{Z}|\mathcal{G};\theta_{enc}) \\
&= -\mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log q(\mathcal{G},\boldsymbol{Z};\theta_{dec}) + \mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log p(\boldsymbol{Z}|\mathcal{G};\theta_{enc}) \\
&= -\mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log q(\mathcal{G}|\boldsymbol{Z};\theta_{dec}) - \mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log p_\theta(\boldsymbol{Z}) + \mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log p(\boldsymbol{Z}|\mathcal{G};\theta_{enc}) \\
&= -\mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log q(\mathcal{G}|\boldsymbol{Z};\theta_{dec}) + D_{\mathrm{KL}}(p(\boldsymbol{Z}|\mathcal{G})||p_\theta(\boldsymbol{Z})) \\
&= \mathcal{L}_{rec} + \mathcal{L}_{diff}
\end{aligned} \tag{18}$$

The second term in $\mathcal{L}_{diff}$ is the negative entropy of the latent variable and has a simply expression w.r.t. the posterior variance:

$$\mathbb{E}_{p(\boldsymbol{Z}|\mathcal{G};\theta_{enc})} \log p(\boldsymbol{Z}|\mathcal{G};\theta_{enc}) = -H(\boldsymbol{\mu}(\mathcal{G};\theta_{enc}) + \boldsymbol{\Sigma}^{\frac{1}{2}}(\mathcal{G};\theta_{enc})\mathcal{E}), \tag{19}$$

with $\mathcal{E} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{NF\times NF})$. For simplicity, we drop $\mathcal{G}, \theta_{enc}$ and omit the constant that is irrelevant of the model, we have

$$\begin{aligned}
H(\boldsymbol{\mu} + \boldsymbol{\Sigma}\mathcal{E}) &= \frac{\log|\boldsymbol{\Sigma}|}{2} - \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \\
&= \frac{\log|\boldsymbol{\Sigma}|}{2} - \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})} Tr((\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})) \\
&= \frac{\log|\boldsymbol{\Sigma}|}{2} - \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})} Tr(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T) \\
&= \frac{\log|\boldsymbol{\Sigma}|}{2} - \frac{1}{2} Tr(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T) \\
&= \frac{\log|\boldsymbol{\Sigma}|}{2}
\end{aligned} \tag{20}$$

Using Eq.9 as the first term in $\mathcal{L}_{diff}$, thus one can efficiently estimate $\mathcal{L}_{diff}$.

A.2 INTRACTABILITY OF CLASSICAL SCORE-MATCHING OBJECTIVES

We demonstrates the intractability of classical score-matching objectives. We can not directly optimize the classic score-matching objectives, due to the intractability of SGM prior $p_\theta(\boldsymbol{Z})$. The explicit score-matching objectives has the form:

$$\mathcal{L}_{ESM} = \mathbb{E}_{t\sim\mathcal{U}(0,T)}\left[\mathbb{E}_{q(\boldsymbol{Z}_t)}[||s_\theta(\boldsymbol{Z}_t,t) - \nabla\log q_t(\boldsymbol{Z}_t)||_2^2]\right], \tag{21}$$

with the estimation $s_\theta(\boldsymbol{Z}_t, t)$. But $\nabla\log q_t(\boldsymbol{Z}_t)$ depends on $p_\theta(\boldsymbol{Z}_0)$ and thus intractable:

$$q_\theta(\boldsymbol{Z}_t) = \int q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)q_\theta(\boldsymbol{Z}_0)d\boldsymbol{Z}_0. \tag{22}$$

One can develop the connection between explicit score-matching objective with the denosing score-matching objective which is the standard objective in SGMs, and we provide a detailed derivation here:

$$\begin{aligned}
\mathcal{L}_{ESM} &= \mathbb{E}_{t\sim\mathcal{U}(0,T)}\left[\mathbb{E}_{q(\boldsymbol{Z}_t)}||s_\theta(\boldsymbol{Z}_t,t)||_2^2 - 2\mathbb{E}_{q(\boldsymbol{Z}_t)}\langle s_\theta(\boldsymbol{Z}_t,t), \nabla\log q_t(\boldsymbol{Z}_t)\rangle\right] + C_1 \\
&= \mathbb{E}_{t\sim\mathcal{U}(0,T)}\left[||s_\theta(\boldsymbol{Z}_t,t)||_{L_2}^2 - 2\mathbb{E}_{q(\boldsymbol{Z}_t)}\langle s_\theta(\boldsymbol{Z}_t,t), \frac{\nabla q_t(\boldsymbol{Z}_t)}{q_t(\boldsymbol{Z}_t)}\rangle\right] + C_1,
\end{aligned} \tag{23}$$

where

$$C_1 = \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{q(\boldsymbol{Z}_t)}||\frac{\nabla q_t(\boldsymbol{Z}_t)}{q_t(\boldsymbol{Z}_t)}||_2^2 \tag{24}$$

The latent space is assumed to be continuous with positive density function, and thus:

$$\begin{aligned}
\mathbb{E}_{q(\boldsymbol{Z}_t)}\langle s_\theta(\boldsymbol{Z}_t,t), \frac{\nabla_{\boldsymbol{Z}_t}\cdot q_t(\boldsymbol{Z}_t)}{q_t(\boldsymbol{Z}_t)}\rangle &= \int \langle s_\theta(\boldsymbol{Z}_t,t), \nabla_{\boldsymbol{Z}_t}\cdot q_t(\boldsymbol{Z}_t)\rangle d\boldsymbol{Z}_t \\
&= \int \langle s_\theta(\boldsymbol{Z}_t,t), \nabla_{\boldsymbol{Z}_t}\cdot \int q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)q(\boldsymbol{Z}_0)d\boldsymbol{Z}_0\rangle d\boldsymbol{Z}_t \\
&= \int \langle s_\theta(\boldsymbol{Z}_t,t), \int \nabla_{\boldsymbol{Z}_t}\cdot q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)q(\boldsymbol{Z}_0)d\boldsymbol{Z}_0\rangle d\boldsymbol{Z}_t \\
&= \int\int \langle s_\theta(\boldsymbol{Z}_t,t), \nabla_{\boldsymbol{Z}_t}\cdot q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)q(\boldsymbol{Z}_0)\rangle d\boldsymbol{Z}_0 d\boldsymbol{Z}_t \\
&= \int\int \langle s_\theta(\boldsymbol{Z}_t,t), q(\boldsymbol{Z}_0)q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)\nabla_{\boldsymbol{Z}_t}\cdot \log q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)\rangle d\boldsymbol{Z}_0 d\boldsymbol{Z}_t \\
&= \int\int q(\boldsymbol{Z}_0)q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)\langle s_\theta(\boldsymbol{Z}_t,t), \nabla_{\boldsymbol{Z}_t}\cdot \log q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)\rangle d\boldsymbol{Z}_t d\boldsymbol{Z}_0 \\
&= \mathbb{E}_{q(\boldsymbol{Z}_0)}\mathbb{E}_{q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)}\langle s_\theta(\boldsymbol{Z}_t,t), \nabla_{\boldsymbol{Z}_t}\cdot \log q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)\rangle.
\end{aligned} \tag{25}$$

Plugging the Eq.25 into $\mathcal{L}_{ESM}$, we have:

$$\begin{aligned}
\mathcal{L}_{ESM} &= \mathbb{E}_{t\sim\mathcal{U}(0,T)}\left[||s_\theta(\boldsymbol{Z}_t,t)||^2_{L_2} - 2\mathbb{E}_{q(\boldsymbol{Z}_0)}\mathbb{E}_{q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)}\langle s_\theta(\boldsymbol{Z}_t,t), \nabla_{\boldsymbol{Z}_t}\cdot \log q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)\rangle\right] + C_1 \\
&= \mathbb{E}_{t\sim\mathcal{U}(0,T)}\left[\mathbb{E}_{q(\boldsymbol{Z}_t,\boldsymbol{Z}_0)}||s_\theta(\boldsymbol{Z}_t,t) - \nabla_{\boldsymbol{Z}_t}\cdot \log q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)||^2_2\right] + C_2 \\
&= \mathcal{L}_{DSM} + C_2
\end{aligned} \tag{26}$$

As a result, we can calculate the difference between explicit score-matching and denosing score-matching objectives.

$$C_2 = \mathbb{E}||\nabla \log q_t(\boldsymbol{Z}_t)||^2_2 - \mathbb{E}||\nabla_{\boldsymbol{Z}_t}\cdot \log q_t(\boldsymbol{Z}_t|\boldsymbol{Z}_0)||^2_2 \tag{27}$$

If one directly diffuse the observed data with prefixed diffusion coefficient, then C_2 is indeed a constant. But if the diffusion is performed on the learned latent space with prior $p_\theta(\boldsymbol{Z}_0)$, then C_2 will depend on the learnable parameter θ, even if the forward diffusion is fixed:

$$q_\theta(\boldsymbol{Z}_t) = \int q(\boldsymbol{Z}_t|\boldsymbol{Z}_0)q_\theta(\boldsymbol{Z}_0)d\boldsymbol{Z}_0 \tag{28}$$

and thus the denosing score matching is also intractable. But the objectives we use, as Eq.9, do not include the term $q_\theta(\boldsymbol{Z}_t)$ and thus tractable.

A.3 Detailed Explanation of Self-Guidance

Here we present a continuous-time setting of self-guidance mechanism: After embedding, we have latent $\boldsymbol{Z}_0$ with pseudo label $\boldsymbol{c}$, with joint distribution $p(\boldsymbol{Z},\boldsymbol{c})$. In the forward SDE, only the latent $\boldsymbol{Z}$ is diffused:

$$d\boldsymbol{Z} = f(t)\boldsymbol{Z}dt + g(t)d\mathbf{w}, \quad \boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{c}) \tag{29}$$

By designing the drift coefficient $f(t)$, the resulting noise $\boldsymbol{Z}_T$ is determined entirely by a independent Brownian motion and forget the initial distribution $\boldsymbol{Z}_0$. For example, we can design:

$$\int_0^T f(s)ds = \infty \tag{30}$$

So the reverse process start with a known prior with label $\boldsymbol{c}$. To reverse the process, we need the reverse SDE with the following form:

$$d\hat{\boldsymbol{Z}} = (f(t)\hat{\boldsymbol{Z}} - g(t)^2\nabla_{\hat{\boldsymbol{Z}}}\log q_t(\hat{\boldsymbol{Z}}|\boldsymbol{c}))dt + g(t)d\bar{\mathbf{w}}. \tag{31}$$

In other word, we need to estimate the conditional score function $\nabla_{\hat{\boldsymbol{Z}}}\log q_t(\hat{\boldsymbol{Z}}|\boldsymbol{c})$ to generate sample with label $\boldsymbol{c}$. Using probability chain rule, one can decompose the conditional score function as:

$$\nabla_{\hat{\boldsymbol{Z}}}\log q_t(\hat{\boldsymbol{Z}}|\boldsymbol{c}) = \nabla_{\hat{\boldsymbol{Z}}}\log q_t(\boldsymbol{c}|\hat{\boldsymbol{Z}}) - \nabla_{\hat{\boldsymbol{Z}}}\log q_t(\hat{\boldsymbol{Z}}) \tag{32}$$

and estimate each part. Real-world graph sets usually do not have explicit label, and thus our model leverages pseudo label to improve the sample quality and cover the whole distribution of target unlabeled graph set by performing self-guidance, as described in section 3.2.

B ADDITIONAL DETAILS AND EXPERIMENTS

B.1 IMPLEMENTATION DETAILS AND HYPERPARAMETERS

Implementation Details of Generic Graph Generation In generic graph generation tasks, we follow the standard setting of existing works (You et al., 2018; Liu et al., 2019; Niu et al., 2020b; Jo et al., 2022). Specifically, we report the result means of 30 runs, 3 different runs for 10 independently trained models on Ego-small and Community-small datasets. And due to the costly training procedure of GraphVAE and EDP-GNN, we report the result means of 3 different runs on Enzymes and Grid datasets. As for the baseline methods implementation, we follow the hyperparameter settings that provided by original works. For SGGM with SLD, we first initilize the node features with the one-hot embedding of the degrees, and then pretrain a variational graph autoencoder. The pretrained autoencoder will initialize a hierarchical pseudo label set by iteratively clustering on all the latent embeddings of the training dataset. The number of classes and hierarchies are detailed in Tab.4 and the hierarchical psuedo label set will be further utilized in SGGM for self-guided diffusion generation with periodic update. We perform the grid search to choose the proper numbers for the hierarchies and classes. After we train SGGM with SLD, we the best MMD with the lowest average of three graph statistics, degree, clustering coefficient, and orbit. For the datasets with small graphs, we use two-layer graph encoder and decoder and a two-layer convolution for score matching while they are of more layers for modeling large graphs, please refer to Tab.4 for further details.

Implementation Details of Molecular Graph Generation In the experiments of molecular graph generation, each molecule is represented by a graph with the node features $\boldsymbol{X} \in \{0,1\}^{N\times F}$ and the adjacency matrix $\boldsymbol{A} \in \{0,1,2,3\}^{N\times N}$, where N denotes the number of atoms in the molecule, and F denotes the number of atom types. The entries of $\boldsymbol{A}$ represent the bond types, i.e. single, double, or triple bonds. Following the standard process of existing works (Shi et al., 2020; Luo et al., 2021b; Jo et al., 2022), the molecules are kekulized by the RDKit library Landrum et al. (2016) and hydrogen atoms are deleted. The whole training pipeline of molecular graph generation is similar to that of generic graph generation, which has been introduced in last paragraph. For further detailed hyperpameter settings, please refer to Tab.4. The novelty value can influence the FCD and NSPDK MMD values. With respect to the evaluation, we choose the hyperparameters that exhibit the best FCD value among those which show the novelty that exceeds 85%. As demonstrated in section 4, we conduct the experiments for extensibility analysis. The geometrical and torsional diffusion processes that used in experiments are implemented according to the published code of their original works (Xu et al., 2021b; Jing et al., 2022).

Table 4: We provide hyperparameters of SGGM with SLD that used in the generic graph generation tasks and the molecule generation tasks.

	Hyperparameter	Ego-small	Community-small	Enzymes	Grid	QM9	ZINC250k
s_θ	Number of convolutional layers	2	3	5	5	2	2
	Hidden dimension	32	32	32	32	16	16
f_{enc}	Number of graph layers	2	2	3	3	2	2
	Hidden dimension	32	32	32	32	16	16
f_{dec}	Number of graph layers	2	2	3	3	2	2
	Hidden dimension	32	32	32	32	16	16
Hierachical label	Number of hierarchies	2	2	3	3	3	3
	Number of update epochs	50	50	50	50	5	5
	Number of classes	{6, 2}	{6, 2}	{12, 4, 2}	{12, 4, 2}	{12, 4, 2}	{12, 4, 2}
Loss Objective	Weight α	1	1	1	1	1	1
	Weight β	0.5	0.5	0.5	0.5	0.5	0.5
Train	Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
	Learning rate	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-2}	5×10^{-3}	5×10^{-3}
	Weight decay	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
	Batch size	128	128	64	8	1024	1024
	Number of epochs	5000	5000	5000	5000	300	500
	Number of sampling steps	600	600	600	600	600	600

B.2 ADDITIONAL EXPERIMENTAL RESULTS

In this subsection, we provide additional experimental results for illustration: generic and molecular graph generation results with the standard deviation; visualization of small, middle, and large graphs generation results.

Table 5: We report the MMD distance between the test datasets and generated graphs with the standard deviation on the Ego-small and the Community-small datasets.

	Ego-small			Community-small		
	Real, $4 \leq \|V\| \leq 18$			Synthetic, $12 \leq \|V\| \leq 20$		
	Deg.	Clus.	Orbit	Deg.	Clus.	Orbit
SGGM	0.025 ± 0.005	0.028 ± 0.004	0.009 ± 0.005	0.041 ± 0.015	0.079 ± 0.134	0.010 ± 0.003
SGGM+SLD	$\mathbf{0.014} \pm 0.006$	$\mathbf{0.019} \pm 0.006$	0.007 ± 0.004	$\mathbf{0.035} \pm 0.017$	$\mathbf{0.071} \pm 0.015$	$\mathbf{0.006} \pm 0.003$

Table 6: We report the MMD distance between the test datasets and generated graphs with the standard deviation on the Enzymes and the Grid datasets.

	Enzymes			Grid		
	Real, $10 \leq \|V\| \leq 125$			Synthetic, $100 \leq \|V\| \leq 400$		
	Deg.	Clus.	Orbit	Deg.	Clus.	Orbit
GraphRNN	$\mathbf{0.017} \pm 0.007$	0.062 ± 0.020	0.046 ± 0.031	$\mathbf{0.064} \pm 0.017$	0.043 ± 0.022	$\mathbf{0.021} \pm 0.007$
GraphAF	1.669 ± 0.024	1.283 ± 0.019	0.266 ± 0.007	-	-	-
GraphDF	1.503 ± 0.011	1.061 ± 0.011	0.202 ± 0.002	-	-	-
GraphVAE	1.369 ± 0.020	0.629 ± 0.005	0.191 ± 0.020	1.619 ± 0.007	0.0 ± 0.000	0.919 ± 0.002
EDP-GNN	0.023 ± 0.012	0.268 ± 0.164	0.082 ± 0.078	0.455 ± 0.319	0.238 ± 0.380	0.328 ± 0.278
GDSS	0.026 ± 0.008	$\mathbf{0.061} \pm 0.010$	0.009 ± 0.005	0.111 ± 0.012	0.005 ± 0.000	0.070 ± 0.044
SGGM	0.030 ± 0.005	0.073 ± 0.008	0.013 ± 0.005	0.114 ± 0.010	$\mathbf{0.0} \pm 0.0$	0.065 ± 0.024
SGGM+SLD	0.022 ± 0.004	$\mathbf{0.062} \pm 0.006$	$\mathbf{0.007} \pm 0.002$	0.103 ± 0.006	$\mathbf{0.0} \pm 0.0$	0.053 ± 0.014

Table 7: Graph generation results are the means and the standard deviations of 3 runs on the QM9 dataset. Validity is the fraction of the generated molecules that do not violate the chemical valency rule. Uniqueness is the fraction of the valid molecules that are unique. Novelty is the fraction of the valid molecules that are not included in the training set.

Method	Validity w/o correction (%) ↑	NSPDK MMD ↓	FCD ↓	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑
GraphAF	67	0.020±0.003	5.268±0.403	**100.00**	94.51	88.83
GraphDF	82.67	0.063±0.001	10.816±0.020	**100.00**	97.62	**98.10**
MoFlow	91.36±1.23	0.017±0.003	4.467±0.595	**100.00**±0.00	98.65±0.57	94.72±0.77
EDP-GNN	47.52±3.60	0.005±0.001	2.680±0.221	**100.00**±0.00	99.25±0.05	86.58±1.85
GraphEBM	8.22±2.24	0.030±0.004	6.143±0.411	**100.00**±0.00	97.90±0.14	97.01±0.17
GDSS	95.72±1.94	**0.003**±0.000	2.900±0.282	**100.00**±0.00	98.46±0.61	86.27±2.29
SGGM	95.91±1.73	0.006±0.001	2.745±0.264	**100.00**±0.00	98.52±0.15	96.61±1.75
SGGM+SLD	**97.35**±1.21	**0.004**±0.000	**2.593**±0.191	**100.00**±0.00	**99.41**±0.11	97.49±1.32

Table 8: Graph generation results are the means and the standard deviations of 3 runs on the ZINC250k dataset. Validity is the fraction of the generated molecules that do not violate the chemical valency rule. Uniqueness is the fraction of the valid molecules that are unique. Novelty is the fraction of the valid molecules that are not included in the training set.

Method	Validity w/o correction (%) ↑	NSPDK MMD ↓	FCD ↓	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑
GraphAF	68	0.044±0.006	16.289±0.482	**100.00**	99.10	**100.00**
GraphAF+FC	68.47±0.99	0.044±0.005	16.023±0.451	**100.00**±0.00	98.64±0.69	99.99±0.01
GraphDF	89.03	0.176±0.001	34.202±0.160	**100.00**	99.16	**100.00**
GraphDF+FC	90.61±4.30	0.177±0.001	33.546±0.150	**100.00**±0.00	99.63±0.01	**100.00**±0.00
MoFlow	63.11±5.17	0.046±0.002	20.931±0.184	**100.00**±0.00	**99.99**±0.01	**100.00**±0.00
EDP-GNN	82.97±2.73	0.049±0.006	16.737±1.300	**100.00**±0.00	99.79±0.08	**100.00**±0.00
GraphEBM	5.29±3.83	0.212±0.075	35.471±5.331	99.96±0.02	98.79±0.15	**100.00**±0.00
GDSS	97.01±0.77	0.019±0.001	14.656±0.680	**100.00**±0.00	99.64±0.13	**100.00**±0.00
SGGM	97.28±0.78	0.018±0.001	13.931±0.625	**100.00**±0.00	98.87±0.14	**100.00**±0.00
SGGM+SLD	**98.32**±0.71	**0.014**±0.001	**11.379**±0.597	**100.00**±0.00	99.83±0.11	**100.00**±0.00

C GENERATION OF SMALL, MIDDLE AND LARGE GRAPHS

We visualize the graphs of small, middle, and large scales from the training datasets and the generated graphs of SGGM with SLD for each dataset in Fig.3-5. We randomly choose samples from the training datasets and the generated graph set for visualization, with e denotes edges number and n denotes nodes number.

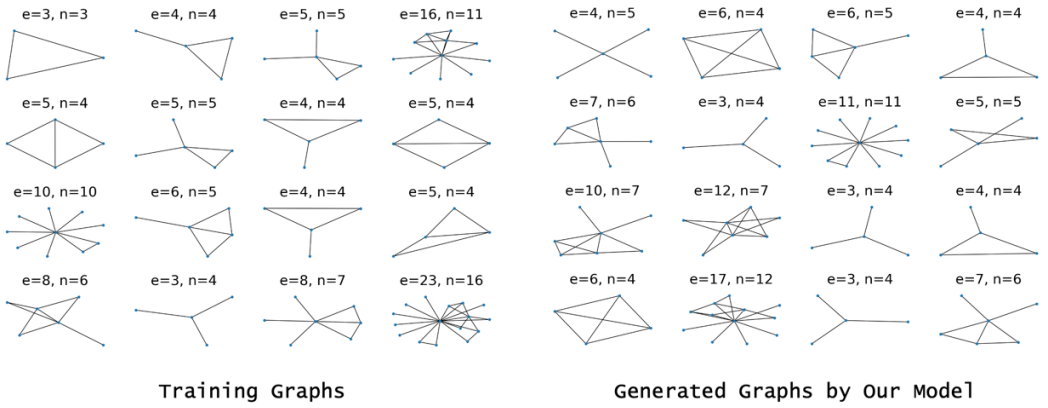

Figure 3: Visualization of the graphs sampled from the Ego-small dataset and the generated graphs of SGGM with SLD.

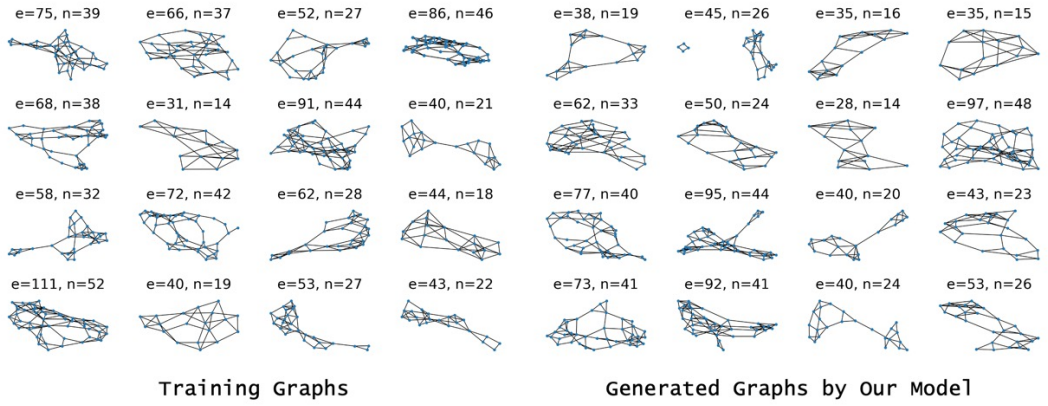

Figure 4: Visualization of the graphs sampled from the ENZYMES dataset and the generated graphs of SGGM with SLD.

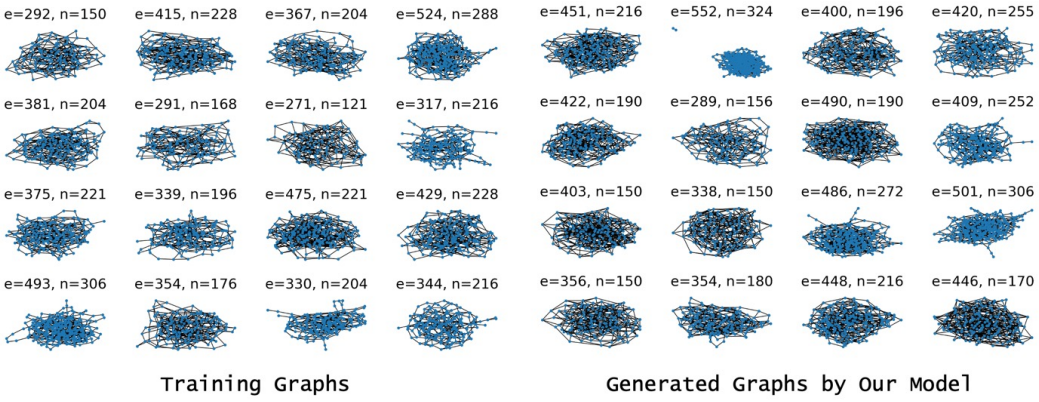

Figure 5: Visualization of the graphs sampled from the Grid dataset and the generated graphs of SGGM with SLD.