A Prior-based Discrete Diffusion Model for Social Graph Generation

Shu Yin^{1,2} , **Dongpeng Hou**^{3,2} , **Lianwei Wu**^{1} , **Xianghua Li**^{2*} and **Chao Gao**^{2}
[1]School of Computer Science, Northwestern Polytechnical University
[2]School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University
[3]School of Mechanical Engineering, Northwestern Polytechnical University
li_xianghua@nwpu.edu.cn

Abstract

Graph generation is essential in social network analysis, particularly for modeling information flow and user interactions. However, existing probabilistic diffusion models face challenges when applied to social propagation graphs. The continuous noise does not apply to the discrete nature of graph generation tasks, and the random Gaussian initialization in the reverse process can introduce biases that deviate from real-world propagation patterns. To address these issues, this paper introduces a Prior-based Discrete Diffusion Model (PDDM) for social graph generation. PDDM redefines the forward process as a discrete process for node denoising and edge generation, and the task of the denoising module is transformed into the connection probability learning of node-level tasks. Further, PDDM employs a new starting point of the reverse process by incorporating user similarity as the probability matrix, which can better leverage the social context. These developments mitigate reverse-starting bias and enhance model robustness. Moreover, PDDM integrates lightweight deep graph networks such as GAT, demonstrating both scalability and applicability to graph generation scenarios. Comprehensive experiments on real-world social network datasets demonstrate PDDM's superiority in terms of the MMD metric and downstream tasks. The code is available at https://github.com/cgao-comp/PDDM.

1 Introduction

Generating graphs based on a target distribution is a fundamental problem in various domains [Liu *et al.*, 2023a]. In the context of social network propagation [Vosoughi *et al.*, 2018], graph generation plays a pivotal role in uncovering hidden patterns of information flow and influence spread, which are crucial for tasks like source localization [Xu *et al.*, 2024a] and user behavior prediction [Zhou *et al.*, 2020]. Understanding the graph generation process in social network propagation sets the foundation for subsequent downstream tasks.

Recent advancements in deep generative models have significantly improved social network analysis, especially compared to traditional random graph models [Leskovec and Faloutsos, 2007]. Deep models, such as Variational Autoencoders (VAEs) [Kingma, 2013], Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014], diffusion model [Ho *et al.*, 2020], and other deep generative approaches, learn to capture complex structural patterns in graph data and generate new graphs with desired properties.

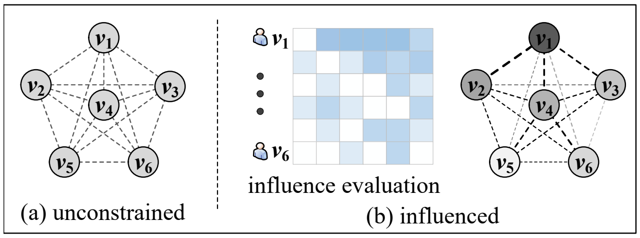


Figure 1: Comparison of unconstrained and influenced graph connectivity. (a) Unconstrained graph connectivity: The graph is fully connected without prior knowledge, leading to a dense but unstructured network. (b) Influenced graph connectivity: Edge formation is guided by influence evaluation, utilizing user similarity to introduce probability into the graph, which ensures more practical connections and enhances the overall quality of the graph generation.

Among them, probabilistic diffusion models have received widespread attention due to their unique ability to handle noise through forward and reverse processes. This noise-based framework allows for flexible customization in learning the complex distributions of social propagation using various social context information, such as user profiles [Jiang *et al.*, 2023], and the propagation structure [Zhu *et al.*, 2024]. Consequently, diffusion models are extensively researched and applied in the generation of propagation graphs within social networks [Cao *et al.*, 2024].

However, existing diffusion models present certain challenges when applied to graph data. Firstly, the forward process in traditional probabilistic diffusion models is inherently continuous, which limits their effectiveness in capturing the discrete nature of graph structures in social networks. Secondly, the reverse process of probabilistic diffusion models typically begins with Gaussian noise, and the iterative reconstruction process often fails to fully eliminate the inherent error, leading to the persistence of bias. Such a biased distribution may not align well with the true underlying distributions of social network graphs.

To overcome these limitations, we propose a Prior based Discrete Diffusion Model (PDDM) for social graph generation. Firstly, PDDM redefines the forward process to adhere to a discrete Markovian process, in which the denoising module learns the node sequence. Secondly, PDDM establishes a new starting point for the reverse process by utilizing user similarity as the probability matrix to mitigate reverse starting bias. These improvements for forward and reverse processes enhance the flexibility and realism of diffusion models for social graph generation. Moreover, it is worth noting that PDDM ensures that the denoising module focuses on node-level tasks, specifically the edge probabilities between new nodes and previously denoised nodes. This enables the convenient use of lightweight models, such as Graph Attention Network (GAT), within the framework, demonstrating the model's generalizability and scalability. The main contributions are as follows.

- To address the limitations of continuous forward processes in traditional diffusion models for the discrete graph generation task, we propose a discrete Markovian forward process. This reformulation enables the denoising module to effectively learn the connection probabilities between new and previously denoised nodes, making the model better suited for capturing discrete structure characteristics.
- To mitigate the bias introduced by Gaussian noise in the reverse process, we introduce a novel reverse process starting point based on user similarity as the probability matrix. The prior guidance of social context ensures that the reverse denoising process more accurately reflects the true structure of social networks, decreasing the bias influence from the random noise.
- PDDM allows for the easy integration of lightweight deep graph models, demonstrating the flexibility and applicability to a wide range of graph generation tasks.

2 Related Work

Graph generation is the task of generating graph-structured data that can serve various applications, including molecular design, social network modeling, and recommendation systems [You *et al.*, 2018a; Jin *et al.*, 2018]. With the advancement of deep learning, researchers have employed deep generative models (such as variational autoencoders, generative adversarial networks, and autoregressive models) for graph data generation [Faez *et al.*, 2021; You *et al.*, 2018b]. For instance, GraphVAE employs the encoder to map a graph to a latent space and the decoder to reconstruct the graph from this representation [Simonovsky and Komodakis, 2018]. This allows for the generation of novel graph instances by sampling from the learned latent space. MolGAN combines graph structures with generative adversarial networks to generate molecular graphs [De Cao and Kipf, 2018]. DAVA adopts a variational autoencoder with exponential distribution sampling for graph-level generation and focuses on user attributes for edge-level generation, allowing for a more nuanced approach to capture user-driven dynamics in the social graphs [Hou *et al.*, 2024].

Denoising diffusion probabilistic models have demonstrated significant potential in various generation fields such as image and video [Ho *et al.*, 2020]. Compared to the aforementioned methods, diffusion-based graph generative models are capable of modeling complex dependencies and generating diverse graph structures [Liu *et al.*, 2023b]. Jo et al. introduce a continuous-time generative model, using stochastic differential equations to model the joint distribution of nodes and edges in the graph diffusion process [Jo *et al.*, 2022]. However, this continuous encoding disrupts the sparsity of the graph. Therefore, DiGress gradually adds noise through independent graph edits (addition/deletion/modification) and trains a graph transformer network to learn the noise process, simplifying the learning of graph structure data distribution into a general classification task [Vignac *et al.*, 2023]. Further, Kong et al. propose an autoregressive diffusion model for graph generation, defining absorbing node states on discrete graphs. In the forward process, each step involves a node autoregressively decaying to the absorbing state, while in the reverse process, a denoising network learns the reverse node absorption diffusion process to reconstruct the graph structure [Kong *et al.*, 2023]. However, the reverse process in these models usually starts with Gaussian noise, and the iterative reconstruction often fails to completely eliminate the inherent errors, resulting in the persistence of bias. This biased distribution may not accurately match the true underlying distributions of social network graphs.

3 Method

This section presents the prior based discrete diffusion model for graph generation. First, we introduce the problem definition and a unified strategy for the graph automorphism. Then, we describe the modifications made to the probabilistic diffusion model for a discrete closed-form solution of social graph generation.

3.1 Problem Definition

Consider a directed acyclic graph $G = (V, E, F)$, which represents a propagation graph from social platforms. In this graph, V denotes the set of users, $E = \{(v_i, v_j)\}$ represents the propagation paths, where each directed edge from v_i to v_j indicates that information flows from user v_i to user v_j. The feature set F contains user-specific attributes, such as profiling and behaviors. In our setting, F includes description, blue verification status, location, registration date, number of posts, number of fans, and number of followings.

The objective is to design a generative model that can generate a new network $G' = (V', E', F')$, where V', E', and F' are the sets of users, propagation paths, and user features in the newly generated network. The generated network G' should exhibit statistical characteristics that closely match those of the observed network G.

3.2 Graph Automorphism

Existing diffusion-based generative models assign a unique ID to each node in the initial graph in order to obtain the unique decay ordering [Chen *et al.*, 2021; Kong *et al.*, 2023].

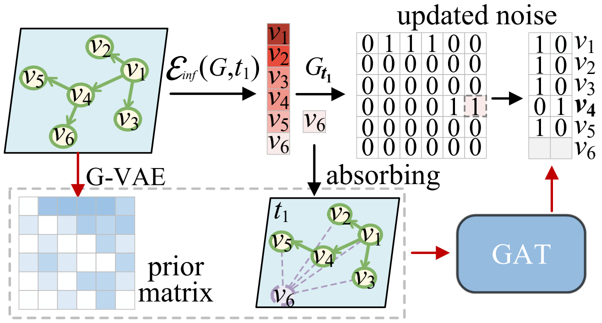
However, this strategy could limit reusability across different graphs. Therefore, a unified influence evaluation strategy based on user profiles is employed.

The six categories of user profiles are collected into $F \in \mathbb{R}^{|V| \times 7}$. A label vector $\mathbf{y}$ of length $|V|$ is constructed, where each entry indicates whether the corresponding cascade is rumor-associated or non-rumor-associated. To avoid scale biases in the chi-square test, we normalize F to F' using Min-Max normalization, ensuring each feature is within the [0,1] range [Patro and Sahu, 2015]. This enables a fair comparison of feature importance. Next, we perform the Chi-Squared test on F' and $\mathbf{y}$ to compute the Chi-Squared statistic for each feature. A smaller p-value indicates a stronger association between the feature and cascade classification. The Chi-Squared statistic for the j-th feature is defined as [Moore, 2017]:

$$\chi^2_{f_k} = \sum_c \sum_v \left(\frac{(f'_k(c,v) - \mu^0_{f_k})^2}{\mu^0_{f_k}} + \frac{((f'_k(c,v) - \mu^1_{f_k})^2}{\mu^1_{f_k}} \right), \tag{1}$$

where $f'_k(c,v)$ is the k-th normalized feature of the user v in the c-th cascade, and $\mu^0_{f_k}$ and $\mu^1_{f_k}$ are the expected values for all non-rumor and rumor-associated cascades, respectively. After sorting the importance of different features $\{f'_1, f'_2, ...\}$, a unique one-hot encoding representation $I(v)$ for each user based on the sorted dimension$\{f'_1, f'_2, ...\}$ can be obtained.

3.3 Discrete Forward Process of PDDM



0	1	1	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	0	0	0
0	0	0	0	0	0

1	0	v_1
1	0	v_2
1	0	v_3
0	1	v_4
1	0	v_5
		v_6

Figure 2: The illustration of the discrete forward process. After addressing the graph automorphism using influence evaluation, the graph is encoded sequentially from the observation state to the noise state based on the node ordering of influence increment. Selected nodes are transitioned to absorbing states and connected to all previously denoised nodes. Subsequently, models such as GNNs are employed to predict the transferred noise, treating it as a discrete node classification problem.

The forward process of the diffusion model is mathematically described by the following equations [Ho *et al.*, 2020]:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right), \tag{2}$$

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}). \tag{3}$$

These equations define the probabilistic progression from any state x_{t-1} to x_t, where β_t are variance parameters that control the noise level at each step.

To eliminate the bias from the random Gaussian noise, we introduce the similarity matrix P as the new starting point to guide the reverse process with the underlying probability of the nodes. Therefore, we use P as a guiding component to adjust the latent distribution space of G, i.e., $\tilde{G}$:

$$\begin{aligned} q(\tilde{G}_t \mid G_{t-1}) = \\ \text{Cat}\left(\tilde{G}_t; \sqrt{1-\beta_t}\,(G_{t-1} - \psi_{t-1}) + (1 - \sqrt{1-\beta_t})P\right), \end{aligned} \tag{4}$$

where P can be conveniently fetched using the variational strategy and attention mechanism. $\tilde{G}_t$ is viewed from a probabilistic perspective, ensuring the smoothness of reverse process and the diversity of the graph generation.

Here, to enhance the applicability of probabilistic diffusion models to the discrete space, we modify the Gaussian noise matrix $\mathbf{I}$. Instead of continuous noise, the variance in graph is transformed into a learnable single interaction matrix ψ_t, corresponding to the propagation edge from node i to node j set to 1. And an operator $\odot$ is defined in a discrete way to handle the perturbations on the graph. However, predicting a single interaction matrix ψ_t with extremely spare topology is infeasible. To address this, we reformulate the problem into a classification task, as shown in Fig. 2. By leveraging GNNs, we can effectively solve this classification problem without the need to directly predict the interaction matrix. Therefore, in this framework, the sparse matrix is utilized solely for deriving a closed-form solution, rather than being directly predicted as noise.

Theorem 1. As $T \to \infty$, $\tilde{G}_T \to P$ in Eq. (4).

Proof: Without loss of generality, we define $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Starting from the iterative diffusion equation:

$$\begin{aligned} \tilde{G}_t &= \sqrt{\alpha_t}\,(G_{t-1} - \psi_{t-1}) + (1 - \sqrt{\alpha_t})P \\ &= \sqrt{\alpha_t \alpha_{t-1}}(G_{t-2} - \psi_{t-1} \odot \psi_{t-2}) + (1 - \sqrt{\alpha_t}\sqrt{\alpha_{t-1}})P \\ &= \cdots \\ &= \sqrt{\bar{\alpha}_t}(G_0 - \overline{\boldsymbol{\psi}}) + (1 - \sqrt{\bar{\alpha}_t})P, \end{aligned}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \psi_{t-1}, \psi_{t-2}, \cdots$ is the single interaction matrix at each step, $\psi_{t-1} \odot \psi_{t-2}$ sums the single interaction matrix cumulatively, $\bar{\boldsymbol{\psi}}_{t-1} = \odot_{i=0}^{t-1} \psi_{t-1}$. $G - \overline{\boldsymbol{\psi}}$ can be interpreted from a discrete perspective, as shown in Fig. 2. Therefore, the operations involving G and graph variance $\overline{\boldsymbol{\psi}}$ are strictly discrete, involving only discrete 0 or 1 updates. In the spatial domain, this corresponds to progressively learning the graph structure by incrementally adding edges as the forward process progresses. As t increases, the cumulative interaction matrix $\overline{\boldsymbol{\psi}}$ gradually approximates G, reflecting the original graph's topology.

As $T \to \infty$, the product $\sqrt{\bar{\alpha}_T}$ tends to zero due to the properties of the parameters α_t which are designed such that $0 \le \alpha_t < 1$. This leads to $\sqrt{\bar{\alpha}_T}(G_0 - \overline{\boldsymbol{\psi}})$ vanishing, and the expression simplifies to:

$$\lim_{T \to \infty} \mathbb{E}[\tilde{G}_T] = (1 - \sqrt{\bar{\alpha}_T})P.$$

$$\lim_{T \to \infty} \text{Var}(\tilde{G}_T) = \sqrt{\bar{\alpha}_t \overline{\boldsymbol{\psi}}}$$

Since $(1-\sqrt{\bar{\alpha}_T}) \to 1$ and $\sqrt{\bar{\alpha}_T} \to 0$ as $T \to \infty$, G_T converges to P. Therefore, $\tilde{G}_T \to P$ as $T \to \infty$. ■

Theorem 2. PDDM ensures a 95% confidence interval for optimal performance when $\min |V| \geq 15$.

Proof: To achieve a 95% confidence interval, the perturbations introduced by the cumulative interaction matrix $\overline{\psi}$ must be constrained within the region around 0, consistent with the properties of standard Gaussian noise, which has a mean of 0. This ensures that the original graph information is retained during the generation process. Consider a propagation graph with $|V|$ nodes, the average edge density is given by:

$$\frac{|V|-1}{|V|^2} < \phi.$$

where ϕ is the threshold derived from the z-value corresponding to the 95% confidence interval, which represents a 95% confidence that the desired threshold ϕ does not deviate far from the mean (i.e., 0). From the CDF of the standard normal distribution, the z-value corresponding to a cumulative probability of 0.525 is approximately $z = 0.063$. Using this z-value as ϕ, we can solve the quadratic inequality and get $|V| \geq 15$. This condition is easily achievable in social networks, which typically involve more than 15 users.

■

Algorithm 1 Training for PDDM

Input: A total of K propagation graphs $G_k = (V_k, E_k, F_k)$.
Output: Optimized parameters θ.

```
repeat
    $V, E, F \sim q(\mathcal{G}_k)$
    Determine the number of the diffusion step $T = |V|$
    Sort user influence $I$ in ascending order based on $F$ using the Chi-Squared statistic
    Deploy a deep module $\theta_1$ to learn similarity matrix $P$
    for $t = 1 \ldots T$ do
        Select the smallest influence node $v_t = I[t]$
        $G(V, E, F) \leftarrow (V \setminus v_t, E \setminus v_t, F \setminus v_t)$
        $G_t \leftarrow \text{Absorb}(G, v_t)$ //Absorb node $v_t$ by connecting it to all nodes in $G$
        $Y_t \leftarrow \{\text{binary\_label}(v_t, w) \mid (w, v) \in E\}$
        Define a denoising module $\hat{Y}_t = \theta_2(P, G_t)$
        Take gradient descent step on: $\nabla_\theta \left\| Y_t - \hat{Y}_t \right\|^2$
    end for
until converged
return optimized parameters $\theta$
```

3.4 Reverse Process of PDDM

As defined in Sec. 3.3, the optimized forward process establishes a discrete framework for graph structures and converges to the prior similarity matrix P. Further, PDDM reconstructs the original graph by iteratively denoising these perturbations. As shown in Fig. 3, the reverse mechanism leverages P to guide the denoising steps, effectively mitigating the bias introduced by random Gaussian noise and ensuring that the generated graph accurately mirrors the underlying social network structure. Similarly to the forward process, by transforming the denoising task into a binary classification problem, the reverse process uses the trained lightweight models to predict edge probabilities.

It is worth mentioning that in the reverse process, directly using the autoregressive step for graph generation is feasible. However, this strategy can lead to issues such as overfitting and a lack of diversity in the generated graphs. In contrast, PDDM employs a closed-form solution based on $q(G_{t-1}|G_t, G)$, leveraging a robust probabilistic foundation. By deriving a discrete version, the reverse process is grounded in a well-defined probabilistic framework, ensuring that the generation process adheres to statistical principles. This results in more diverse graph generation and a closer reflection of real-world social graph distributions.

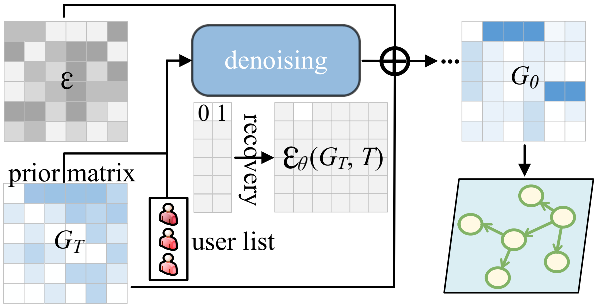


Figure 3: The illustration of the reverse process. The process begins by utilizing a similarity matrix P as a prior, which serves as the starting point for the reverse diffusion. The denoising module evaluates the probabilities of candidate users against all previously denoised nodes. These probabilities are then represented as a single interaction matrix, facilitating closed-form mathematical operations. And at each step of the reverse process, modified smoothing noise is introduced. This addition enhances the stability of the denoising iterations by mitigating discrete influence.

Consistent with the traditional diffusion model [Ho *et al.*, 2020], the variational lower bound (VLB) for PDDM is given by:

$$\mathcal{L}_{\text{VLB}} = \mathbb{E}_{q(G_{0:T})}\left[\log \frac{q(G_{1:T} \mid G_0)}{p_\theta(G'_{0:T})}\right] = \mathbb{E}_q\Big[D_{\text{KL}}\big(q(G_T \mid G_0) \parallel p_\theta(G'_T)\big) + \sum_{t=2}^{T} D_{\text{KL}}\big(q(G_{t-1} \mid G_t, G_0) \parallel p_\theta(G'_{t-1} \mid G'_t)\big) - \log p_\theta(G'_0 \mid G'_1)\Big] \tag{5}$$

where G_0 is the original graph G, and $G_{1:T}$ represents the graph denoising the node in a user influence increment order. Here, $\mathcal{L}_{\text{VLB}}$ serves as the objective function to be maximized during training, ensuring that the generated graph G'_0 closely resembles the original graph G. Among them, the summation $\sum_{t=2}^{T} \text{KL}\left(q(G_{t-1} \mid G_t, G) \| p_\theta(G'_{t-1} \mid G'_t)\right)$ represents the cumulative KL divergence between the true reverse distribution q and the model's reverse distribution p_θ across all diffusion steps. Minimizing this term aligns the model's predictions with the true data distribution. The conditional probability distribution of the reverse process is as follows:

$$
\begin{aligned}
q(G_{t-1} \mid G_t, G) &= \frac{q(G)q(G_{t-1} \mid G)q(G_t \mid G_{t-1}, G)}{q(G)q(G_t \mid G)} \\
&= q(G_t \mid G_{t-1}, G)\frac{q(G_{t-1} \mid G)}{q(G_t \mid G)}.
\end{aligned} \tag{6}
$$

Given that the forward process is a Markov chain, $q(G_t \mid G_{t-1}, G)$ is independent of G, thus we can get:

$$
q(G_{t-1} \mid G_t, G) = q(G_t \mid G_{t-1})\frac{q(G_{t-1} \mid G)}{q(G_t \mid G)}. \tag{7}
$$

Theorem 3. For a $|V| \times |V|$ sparse matrix G with exactly t elements equal to 1 and all other elements equal to 0, and a $|V| \times |V|$ Gaussian matrix X where each element X_{ij} is independently sampled from the standard normal distribution $\mathcal{N}(0, 1)$, scaling X by a factor of $\frac{\sqrt{t}}{|V|}$ ensures that the variance of each element in the scaled matrix $\tilde{X} = \frac{\sqrt{t}}{|V|}X$ matches the variance of the corresponding element in G.

Proof. Consider the sparse matrix $G \in \{0, 1\}^{|V|\times|V|}$ where exactly t elements are 1 and the remaining $|V|^2 - t$ elements are 0. For large $|V|$, the mean μ_G and variance σ_G^2 of each element G_{ij} are approximately:

$$
\mu_G = \mathbb{E}[G_{ij}] = \frac{t}{|V|^2}
$$

$$
\sigma_G^2 = \mathbb{E}[G_{ij}^2] - (\mathbb{E}[G_{ij}])^2 = \frac{t}{|V|^2} - \left(\frac{t}{|V|^2}\right)^2 \approx \frac{t}{|V|^2}
$$

Since the number of nodes in a social network is large, the above equality condition is easily established. Now, consider the Gaussian matrix $X \in \mathbb{R}^{|V|\times|V|}$ where each element $X_{ij} \sim \mathcal{N}(0, 1)$. We can get the variance $\mathrm{Var}(X_{ij}) = 1$.

To match the variance of the sparse matrix G, we scale the Gaussian matrix X by a factor $c = \frac{\sqrt{t}}{|V|}$, resulting in the scaled matrix $\tilde{X} = cX$. The variance of each element in $\tilde{X}$ is then:

$$
\mathrm{Var}(\tilde{X}_{ij}) = \mathrm{Var}\left(\frac{\sqrt{t}}{|V|}X_{ij}\right) = \left(\frac{\sqrt{t}}{|V|}\right)^2 \cdot \mathrm{Var}(X_{ij}) = \frac{t}{|V|^2}
$$

Thus, by setting $c = \frac{\sqrt{t}}{|V|}$, we ensure that:

$$
\mathrm{Var}(\tilde{X}_{ij}) = \frac{t}{|V|^2} = \mathrm{Var}(G_{ij})
$$

Therefore, the scaled Gaussian matrix $\tilde{X}$ has consistent variances that match those of the sparse matrix G. ■

Based on Theorem 1, we can get the three forward equations $q(G_t \mid G_{t-1})$, $q(G_{t-1} \mid G)$, and $q(G_t \mid G)$ in Eq. (7). And through scaling, Theorem 3 ensures that the variance remains consistent across different diffusion steps and guarantees smoothness. Based on the probability density function, Eq. (7) can be further expanded as follows:

$$
\begin{aligned}
&q(G_{t-1}|G_t, G) \propto \exp \\
&\left\{-\frac{1}{2}\left(\left(1+\frac{1}{(t-1)\bar{\alpha}_{t-1}}\right)G_{t-1}^2+\left(\frac{1}{\alpha_t}\left(-2\sqrt{\alpha_t}G_t + 2(\sqrt{\alpha_t} - \alpha_t)P\right) + \right.\right.\right. \\
&\left.\left.\left.\frac{1}{(t-1)\bar{\alpha}_{t-1}}\left(-2\sqrt{\bar{\alpha}_{t-1}}G - 2(1 - \sqrt{\bar{\alpha}_{t-1}})P\right)\right)G_{t-1}+C(G_t, G)\right)\right\}.
\end{aligned} \tag{8}
$$

After the single interaction matrix is evaluated, the perturbation coefficient can be determined as follows:

$$
\sigma_{t-1|t}^2 = \left(1+\frac{1}{(t-1)\bar{\alpha}_{t-1}}\right)^{-1} = \sigma_{t-1|t}^2 = \frac{(t-1)\bar{\alpha}_{t-1}}{(t-1)\bar{\alpha}_{t-1}+1}. \tag{9}
$$

And the mean can be determined as follows:

$$
\begin{aligned}
\mu_{t-1|t} = \sigma_{t-1|t}^2 \cdot &\left(\frac{\sqrt{\alpha_t}}{\alpha_t}G_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{(t-1)\bar{\alpha}_{t-1}}G\right. \\
&\left.-\left(\frac{\sqrt{\alpha_t}}{\alpha_t} - 1 - \frac{(1 - \sqrt{\bar{\alpha}_{t-1}})}{(t-1)\bar{\alpha}_{t-1}}\right)P\right).
\end{aligned} \tag{10}
$$

Based on $q(G_t \mid G) = \sqrt{\bar{\alpha}_t}(G - \sqrt{1 - \bar{\alpha}_t\overline{\boldsymbol{\psi}}_{t-1}}) + (1 - \sqrt{\bar{\alpha}_t})P$, we can get:

$$
G = \frac{G_t - (1 - \sqrt{\bar{\alpha}_t})P}{\sqrt{\bar{\alpha}_t}} + \sqrt{1 - \bar{\alpha}_t\overline{\boldsymbol{\psi}}_{t-1}}. \tag{11}
$$

Then, Eq. (10) for $\mu_{t-1|t}$ can be demonstrated as follows:

$$
\begin{aligned}
\mu_{t-1|t} = &\frac{(t-1)\bar{\alpha}_{t-1}+1}{\sqrt{\alpha_t}(t-1)\bar{\alpha}_{t-1}}G_t + (\frac{\sqrt{\alpha_t}-1}{\sqrt{\alpha_t}} + \frac{\sqrt{\alpha_t}-1}{\sqrt{\alpha_t}(t-1)\bar{\alpha}_{t-1}})P \\
&+ \frac{\sqrt{\bar{\alpha}_{t-1}}}{(t-1)\bar{\alpha}_{t-1}} \cdot \sqrt{1 - \bar{\alpha}_t\overline{\boldsymbol{\psi}}_{t-1}}.
\end{aligned} \tag{12}
$$

Thus, the variance $\sigma_{t-1|t}^2$ of the posterior conditional distribution from a probabilistic perspective is given by Eq. (9), and the mean $\mu_{t-1|t}$ is expressed by Eq. (12). With these closed-form solutions established, we can predict the edge appearance discretely and the graph sampling continuously, which ensures the discreteness required for graph generation tasks while also enhancing the diversity.

The training process of PDDM is demonstrated in Alg. 1. In lines 2–5, the algorithm solves the graph permutation problem based on the user influence evaluation and deploys a deep module θ_1 to learn the similarity matrix P. Then, lines 7–12 iterates over the discrete absorbing process and the denoising module θ_2 predicts the constructed binary label Y_t. This procedure repeats until convergence, ensuring that PDDM learns to handle the discrete structure of social networks while capturing rich propagation characteristics. However, discrete learning for the edge consistently replicates the original graph edge connection distribution. This coarse-grained replication lacks transferability in unseen continuous spaces and cannot guarantee the diversity of the generative capacity.

Therefore, the closed-form reverse diffusion process in Alg. 2 from T to t_1 corrects the coarse-grained replication. After T iterations, the cumulative interaction matrix $\overline{\boldsymbol{\psi}}$ is guided by the prior P from a probabilistic perspective, while the variance introduced in line 7 is correctly scaled according to Theorem 3. Finally, edges with the highest probabilities are selected based on a permutation influence order, reconstructing $\hat{G}$ with diversity by robustly introducing randomness.

4 Experiments

4.1 Datasets and Baselines

We use real-world propagation graphs on Weibo and Twitter platforms for graph generation, namely Weibo [Ma *et al.*,

Algorithm 2 Graph generation process of PDDM

```
Input: The optimized denoising module $\theta$; A candidate user set $(V, F)$.
Output: The generative graph $\tilde{G}$.
Determine the number of the diffusion step $T = |V|$
Sort user influence $I$ in ascending order
$P = \theta_1(V, F), \tilde{G}_t \leftarrow P$
Compute $Y_1 \cdots Y_t$ and construct $\bar{\boldsymbol{\psi}}_{t-1} \cdots \bar{\boldsymbol{\psi}}_0$
for $t = T, \ldots, 1$ do
    $z \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $z = 0$
    $\sigma^2_{t-1|t} = \frac{t}{|V|^2} \cdot \frac{(t-1)\bar{\alpha}_{t-1}}{(t-1)\bar{\alpha}_{t-1}+1}$ //Eq. (9)
    $\mu_{t-1|t} = \frac{(t-1)\bar{\alpha}_{t-1}+1}{\sqrt{\alpha_t}(t-1)\bar{\alpha}_{t-1}}\tilde{G}_t + (\frac{\sqrt{\alpha_t}-1}{\sqrt{\alpha_t}} + \frac{\sqrt{\alpha_t}-1}{\sqrt{\alpha_t}(t-1)\bar{\alpha}_{t-1}})P + \frac{\sqrt{\bar{\alpha}_{t-1}}}{(t-1)\bar{\alpha}_{t-1}} \cdot \sqrt{1-\bar{\alpha}_t}\bar{\boldsymbol{\psi}}_{t-1}$ //Eq. (12)
    $\tilde{G}_{t-1} = \mu_{t-1|t} + \sigma_{t-1|t} \cdot z$
end for
return $\tilde{G} = \mathrm{argmax}_{v \in \mathrm{permutation}}\, \tilde{G}_0(v)$
```

2017], Twitter15, and Twitter16 [Liu *et al.*, 2015; Ma *et al.*, 2016]. Further, based on the user IDs that participated in propagation cascades of the Twitter platform, we collect the user profiles to enrich the individual characteristics of propagation, including user description, blue verification status, location, registration date, number of posts, number of fans, and number of followings. The relevant information of the three datasets is shown in Tab. 1.

Statistic	Twitter15	Twitter16	Weibo
#users	480,987	289,675	2,856,741
#users in $\mathcal{G}$	480,405	289,504	2,856,519
#relations in $\mathcal{G}$	565,948	334,603	3,508,596
#tweets	1490	818	4664

Table 1: Statistics of the datasets. $\mathcal{G}$ is the largest component of the joint historical relationship network based on UIDs.

To highlight the generative performance of the proposed methods, we choose seven SOTA methods for comparison. And we compare the SOTA methods of DAVA [Hou *et al.*, 2024], GRAPHARM [Kong *et al.*, 2023] (denoted as ***ARM***), DAGG [Han *et al.*, 2023], GVAE_MM [Zahirnia *et al.*, 2022], D-VAE [Zhang *et al.*, 2019], GraphVAE [Simonovsky and Komodakis, 2018], GraphRNN [You *et al.*, 2018b].

4.2 Experimental Setting

We use two strategies to evaluate the performance of data generation. (1) Following the SOTA settings, we use 80% of the graphs as training set and the rest 20% as test sets [Kong *et al.*, 2023] for each dataset. We measure generation quality using the maximum mean discrepancy (MMD) [Kawai *et al.*, 2019] as a distribution distance between the generated graphs and the test graphs. We generate the same number of samples as the test set for each dataset, based on the users in each graph of the test set. Specifically, we compute the MMD of degree distribution and the normalized minimum number of graph edit distances required to transform the generated graphs into real-world graphs in the test set. (2) To demonstrate the practicality of the generated data in real-world scenarios, we use source localization as an example to explore whether utilizing generated data can enhance the model's predictive ability in real-world contexts. We also report the generation time of different methods.

4.3 Overall Performance

MMD Based Metric Evaluation

Tab. 2 summarizes the MMD evaluation results for the generated propagation graph across all social platforms. To ensure the generalizability and reusability of the proposed framework, we employ convenient GCNs and GATs as the denoising module. It can be seen that PDDM consistently outperforms existing methods, achieving an average 15.60% reduction in MMD compared to the best baseline across datasets, both in terms of degree distribution and normalized graph edit distances. The improved generation capability of PDDM can be attributed to three main factors. (1) The developed discrete forward diffusion process enhances the learning ability to capture discrete structure characteristics from the perspective of edge connection probability, which provides a suitable strategy for topology generation. (2) The employed new reverse starting point mitigates random Gaussian noise bias, leveraging user similarity as the probability matrix to ensure more accurate structure generation aligned with social context. (3) The application of the reverse diffusion process in a probabilistic distribution space not only corrects the coarse-grained generation but also introduces robustness to enhance the diversity of generated graphs.

Metrics	**Deg.**			**Dis.**			**Time**
Dataset	*T15*	*T16*	*Wb*	*T15*	*T16*	*Wb*	*2000*
DAGG	0.155	0.146	0.122	0.296	0.276	0.288	-
GVAE_MM	0.171	0.183	0.180	0.320	0.281	0.266	0.83
D-VAE	0.214	0.204	0.219	0.282	0.223	0.272	-
GraphVAE	0.225	0.234	0.271	0.358	0.355	0.306	0.33
GraphRNN	0.154	0.147	0.177	0.265	0.247	0.351	0.5
DAVA	0.064	0.066	0.114	0.188	0.170	0.213	**0.08**
ARM	0.077	0.083	0.108	0.201	0.160	0.235	0.25
PDDM	**0.058**	**0.052**	**0.065**	**0.176**	**0.166**	**0.189**	0.16
SD	±0.005	±0.005	±0.008	±0.014	±0.013	±0.018	

Table 2: The generation performance evaluation of different methods based on MMD metric. The time signifies the approximate hours needed for the model to generate a single graph of 2,000 nodes. The bold values represent the best results (the smaller the better).

We further incorporate diversity-related metrics, including *Uniqueness* and *Novelty* [Vignac *et al.*, 2023; Xu *et al.*, 2024b]. Specifically, *Uniqueness* reports the fraction of generated non-isomorphic graphs, and *Novelty* reports the fraction of the generated graphs that are not isomorphic to any graph from the training set. The **Unique** and **Novelty** columns in Table 3 show that PDDM consistently maintains high diversity.

Metrics	**Unique**		**Novelty**	
Dataset	*T15*	*Wb*	*T15*	*Wb*
DAVA	0.98	**1.00**	0.89	0.96
PDDM	**1.00** (±0.00)	**1.00** (±0.00)	**1.00** (±0.00)	**1.00** (±0.00)

Table 3: The generation performance based on the Uniqueness and Novelty metrics.

Utility of Generated Data in Downstream Tasks

Many downstream tasks, such as information cascade prediction, influence maximization, fake news detection, and source localization, rely on propagation models [Hou *et al.*, 2024]. To analyze whether generated data can enhance model performance in downstream tasks, we use source localization as a case study. The convenient localization models GCNSI [Dong *et al.*, 2019] and TGASI [Hou *et al.*, 2023] can easily use the generated snapshots for training and source prediction. In the experiment, we use four groups to ensure the rigor of the study.

- The **original group** trains the localization models on 90% of the Twitter propagation data and tests on the remaining 10%.
- The **augmentation group of PDDM** additionally generates 1,000 propagation graphs using PDDM and adds the generated graph to the training set.
- The **augmentation group of SOTA models** additionally generates 1,000 propagation graphs using DAVA and GRAPHARM for training.
- The **control group** simulates 1,000 snapshots using traditional SI, SIR, IC, and LT models.

As shown in Tab. 4, training on simulated data from traditional models leads to reduced performance on downstream tasks in real-world propagation scenarios, indicating limited applicability of these models to actual tasks. In contrast, augmenting with real generated data results in improved performance, with the best results observed when using propagation data generated by PDDM. This highlights the significance of realistic graph generation and underscores PDDM's effectiveness in enhancing model performance.

Strategy	Original	Augmented (PDDM/SOTA)	Control
GCNSI	0.532	**0.642**/0.625	0.512
TGASI	0.787	**0.855**/0.834	0.755

Table 4: Source detection accuracy of localization methods under different groups of training sets.

4.4 Ablation Study

We further study the effectiveness of the components to verify their contributions to graph generation tasks. The critical ablation settings include:

- "$P \rightarrow N(0, \mathbf{I})$" uses the standard Gaussian noise in the DDPM instead of the user similarity matrix at the T step.
- "Discrete → DDPM" uses the Vanilla DDPM [Ho *et al.*, 2020] instead of the proposed discrete forward process.
- "-Reverse" removes the reverse diffusion process of the probabilistic distribution perspective.
- "-Att" removes the graph attention module, and only the GCNs are available.
- "-PE" does not consider the time step index in the denoising module.

As shown in Tab. 5, it will lead to a performance decrease no matter removing or exchanging any critical modules. Among the changes, the most significant performance drop occurs when the discrete forward process is omitted. This suggests that the discrete forward process plays a pivotal role in capturing the inherent topology characteristics of the graph generation task. Without the discrete process, the model struggles to effectively learn the edge connection distribution, leading to distinct degraded performance. Furthermore, the influence of the diffusion framework is found to be greater than that of the denoising module. This highlights the crucial role of the proposed diffusion framework in ensuring effective graph generation. The results indicate that the PDDM framework benefits most from its novelly designed forward and reverse processes. Additionally, the attention mechanism has the greatest impact on the generation time, as it is involved in every step of the computation, resulting in significant overhead.

Modules	*Variants*	Deg.	Dis.	*Time (h)*
Diffusion	$P \rightarrow N(0, \mathbf{I})$	0.125	0.206	0.14
	Discrete→DDPM	0.161	0.294	0.25
	-Reverse	0.114	0.212	0.1
Denoising	-Att	0.103	0.205	0.08
	-PE	0.077	0.195	0.15
Origin	PDDM	**0.058**	**0.176**	**0.16**

Table 5: The generation performance evaluation of the variant model from PDDM based on MMD metric in Twitter15.

5 Conclusion

In social graph generation tasks, existing probabilistic diffusion models may face challenges such as the limited ability to handle discrete topology characteristics and the introduced bias at the reverse starting point sampling by the random Gaussian. In this paper, we introduce a prior-based discrete diffusion model to address these limitations. By redefining the forward process as a discrete Markovian process, PDDM transfers the random noise learning task to the edge connection probability learning task from a discrete generative perspective of social propagation graphs. Furthermore, by introducing a novel reverse process starting point based on user similarity as the probability matrix, PDDM mitigates reverse starting bias and aligns better with the true structure of social networks. Not only does PDDM outperform in terms of MMD metrics, but the generated data also leads to improvements in downstream tasks, proving the practical significance.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Nos. 62271411, U22A2098, 62261136549, 62471403), the Technological Innovation Team of Shaanxi Province (No. 2025RS-CXTD-009), the International Cooperation Project of Shaanxi Province (No. 2025GH-YBXM-017), the Fundamental Research Funds for the Central Universities (Nos. G2024WD0151, D5000240309).

References

[Cao *et al.*, 2024] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Chen *et al.*, 2021] Xiaohui Chen, Xu Han, Jiajing Hu, Francisco Ruiz, and Liping Liu. Order matters: Probabilistic modeling of node sequence for graph generation. In *International Conference on Machine Learning*, pages 1630–1639, 2021.

[De Cao and Kipf, 2018] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[Dong *et al.*, 2019] Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 569–578, 2019.

[Faez *et al.*, 2021] Faezeh Faez, Yassaman Ommi, Mahdieh Soleymani Baghshah, and Hamid R Rabiee. Deep graph generators: A survey. *IEEE Access*, 9:106675–106702, 2021.

[Goodfellow *et al.*, 2014] 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.

[Han *et al.*, 2023] Xu Han, Xiaohui Chen, Francisco JR Ruiz, and Li-Ping Liu. Fitting autoregressive graph generative models through maximum likelihood estimation. *Journal of Machine Learning Research*, 24(97):1–30, 2023.

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

[Hou *et al.*, 2023] Dongpeng Hou, Zhen Wang, Chao Gao, and Xuelong Li. Sequential attention source identification based on feature representation. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 1–9, 2023.

[Hou *et al.*, 2024] Dongpeng Hou, Chao Gao, Xuelong Li, and Zhen Wang. Dag-aware variational autoencoder for social propagation graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8508–8516, 2024.

[Jiang *et al.*, 2023] Julie Jiang, Xiang Ren, and Emilio Ferrara. Retweet-bert: political leaning detection using language features and information diffusion on social networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 17, pages 459–469, 2023.

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

[Jo *et al.*, 2022] 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*, pages 10362–10383, 2022.

[Kawai *et al.*, 2019] Wataru Kawai, Yusuke Mukuta, and Tatsuya Harada. Scalable generative models for graphs with graph attention mechanism. *arXiv preprint arXiv:1906.01861*, 2019.

[Kingma, 2013] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Kong *et al.*, 2023] Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In *International conference on machine learning*, pages 17391–17408, 2023.

[Leskovec and Faloutsos, 2007] Jure Leskovec and Christos Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *Proceedings of the 24th international conference on Machine learning*, pages 497–504, 2007.

[Liu *et al.*, 2015] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870, 2015.

[Liu *et al.*, 2023a] Chengyi Liu, Wenqi Fan, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. Generative diffusion models on graphs: methods and applications. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6702–6711, 2023.

[Liu *et al.*, 2023b] Chengyi Liu, Wenqi Fan, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. Generative diffusion models on graphs: methods and applications. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6702–6711, 2023.

[Ma *et al.*, 2016] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Cha Meeyoung. Detecting rumors from microblogs with recurrent neural networks. In *The 25th International Joint Conference on Artificial Intelligence*. AAAI, 2016.

[Ma *et al.*, 2017] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 708–717, 2017.

[Moore, 2017] David S Moore. Tests of chi-squared type. In *Goodness-of-fit-techniques*, pages 63–96. Routledge, 2017.

[Patro and Sahu, 2015] SGOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[Simonovsky and Komodakis, 2018] 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*, pages 412–422, 2018.

[Vignac *et al.*, 2023] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023.

[Vosoughi *et al.*, 2018] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *science*, 359(6380):1146–1151, 2018.

[Xu *et al.*, 2024a] Xovee Xu, Tangjiang Qian, Zhe Xiao, Ni Zhang, Jin Wu, and Fan Zhou. Pgsl: A probabilistic graph diffusion model for source localization. *Expert Systems with Applications*, 238:122028, 2024.

[Xu *et al.*, 2024b] Zhe Xu, Ruizhong Qiu, Yuzhong Chen, Huiyuan Chen, Xiran Fan, Menghai Pan, Zhichen Zeng, Mahashweta Das, and Hanghang Tong. Discrete-state continuous-time diffusion for graph generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, pages 1–37, 2024.

[You *et al.*, 2018a] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

[You *et al.*, 2018b] 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*, pages 5708–5717, 2018.

[Zahirnia *et al.*, 2022] Kiarash Zahirnia, Oliver Schulte, Parmis Naddaf, and Ke Li. Micro and macro level graph modeling for graph variational auto-encoders. In *Advances in Neural Information Processing Systems*, volume 35, pages 30347–30361, 2022.

[Zhang *et al.*, 2019] Muhan Zhang, Shali Jiang, Zhicheng Cui, Roman Garnett, and Yixin Chen. D-vae: A variational autoencoder for directed acyclic graphs. In *Advances in Neural Information Processing Systems*, pages 1586–1598, 2019.

[Zhou *et al.*, 2020] Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. A data-driven graph generative model for temporal interaction networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 401–411, 2020.

[Zhu *et al.*, 2024] Junyou Zhu, Chao Gao, Ze Yin, Xianghua Li, and Jürgen Kurths. Propagation structure-aware graph transformer for robust and interpretable fake news detection. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4652–4663, 2024.