CDGRAPH: DUAL CONDITIONAL SOCIAL GRAPH SYNTHESIZING VIA DIFFUSION MODEL

Anonymous authors
Paper under double-blind review

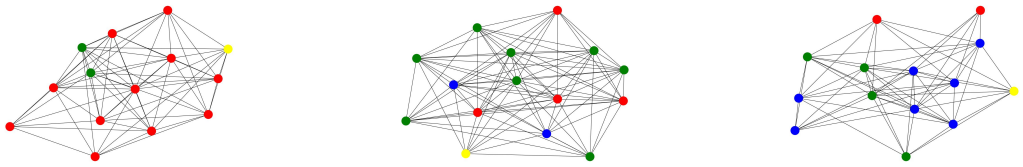
ABSTRACT

The social graphs synthesized by the generative models are increasingly in demand due to data scarcity and concerns over user privacy. One of the key performance criteria for generating social networks is the fidelity to specified conditionals, such as users with certain membership and financial status. While recent diffusion models have shown remarkable performance in generating images, their effectiveness in synthesizing graphs has not yet been explored in the context of conditional social graphs. In this paper, we propose the first kind of conditional diffusion model for social networks, `CDGraph`, which trains and synthesizes graphs based on two specified conditions. We propose the co-evolution dependency in the denoising process of `CDGraph` to capture the mutual dependencies between the dual conditions and further incorporate social homophily and social contagion to preserve the connectivity between nodes while satisfying the specified conditions. Moreover, we introduce a novel classifier loss, which guides the training of the diffusion process through the mutual dependency of dual conditions. We evaluate `CDGraph` against four existing graph generative methods, i.e., SPECTRE, GSM, EDGE, and DiGress, on four datasets. Our results show that the generated graphs from `CDGraph` achieve much higher dual-conditional validity and lower discrepancy in various social network metrics than the baselines, thus demonstrating its proficiency in generating dual-conditional social graphs.

1 INTRODUCTION

Social networks offer a wide range of applications, such as viral marketing, friend recommendations, fake news detection, and more. However, achieving effective results often requires a substantial amount of personal data. Nevertheless, with the rise of privacy awareness, most individuals are reluctant to publicly disclose their personal information, including their profile and social interaction records, leading to a scarcity of data. The need of generating a social graph similar to the original one arises. It is critical for synthetic graphs to not only have similar structures as the original ones, e.g., centrality, but also satisfy exogenous conditions, such as specific user profiles. For example, when evaluating a customer's influence on a luxury golf club brand, it is more relevant to analyze a subset of her friends who have golf as a hobby, rather than analyzing her entire group of friends.

Statistical sampling approaches (Shuai et al., 2018; Schweimer et al., 2022) have been used to produce graphs with certain social network properties, such as skewed degree distribution, a small diameter, and a large connected component, but they struggle to ensure the structure similarity to the original social graphs. Moreover, these methods cannot control the generation process to satisfy specified conditions (e.g., profiles of a social network user) (Bonifati et al., 2020). Recently, deep generative models are shown effective for synthesizing molecular graphs (Samanta et al., 2020; Chenthamarakshan et al., 2020), via extracting latent features from input graphs. The deep molecular graphs (Huang et al., 2022; Vignac et al., 2023) can well preserve the network structure and deal with a single exogenous condition only (e.g., chemical properties such as toxicity, acidity, etc.) but not more. They fail to capture the dependency between two specified conditions, and thus cannot generate graphs satisfying both conditions. For instance, in the above example, (Huang et al., 2022; Vignac et al., 2023) may synthesize the graph satisfying the condition of users being golf enthusiasts, but they do not consider the income bracket relevant to purchasing power.



(a) Unconditional. (b) Single-conditional. (c) Dual-conditional.

Figure 1: Illustrative examples of synthetic social graphs generated by unconditional, single-conditional, and dual-conditional approaches. The colors of the node represent the condition satisfaction: none (red), one (green and yellow), and both (blue).

Generating social graphs satisfying dual conditions is no mean feat and faces several challenges. (i) *Intricate dependencies across conditions*: Those exogenous conditions are often mutually dependent, and modeling conditions independently may lead to sub-optimal synthetic graphs. For instance, there is a high correlation between being golf enthusiasts and income brackets. (ii) *Fulfilling graph structure similarity and exogenous conditions*: When synthesizing social graphs, one has to not only maintain the network structure but also adhere to the conditions. Compared to chemical graphs, where the structure is dictated primarily by physical and chemical constraints, social homophily and social contagion are prevalent phenomena in social graphs. That is, users' profiles subtly drive their social interactions and vice versa. Hence, it is crucial to follow the original structure to generate and link users based on their profiles while ensuring that dual conditions are met (i.e., avoiding generating an excessive number of users who do not meet the specified conditions). Figure 1 illustrates the generated graphs obtained by unconditional, single-conditional, and dual-conditional generative models. It can be observed that both unconditional and single-conditional methods fall short in generating graphs that satisfy dual conditions, all the while maintaining a network structure influenced by social homophily and social contagion.

In this paper, we propose a novel conditional diffusion model, *Dual Conditional Diffusion Graph (`CDGraph`)*, for synthesizing social graphs based on two exogenous conditions jointly. We first propose a novel notion of *co-evolution dependency* to capture the mutual dependencies between two exogenous conditions. In the conditional denoising process, we introduce the *co-evolution dependency* to bind the diffusion processes to the specified node conditions. Moreover, the co-evolution dependency is designed to account for *social homophily* and *social contagion*, which explore the dependencies between the nodes' associated conditions and connections. The Social homophily-based co-evolution ensures nodes with similar profiles are connected, while the social contagion-based co-evolution facilitates nodes connected with edges to share similar profiles. Equipped with the co-evolution dependency, `CDGraph` can preserve the connectivity between nodes that satisfy the specified conditions. Then, we design a novel loss function to train `CDGraph` with the guidance of the specified conditions, aiming to optimize the discrepancy in the co-evolution diffusion process. Furthermore, we introduce the notion of the *dual-condition classifier* that jointly steers the co-evolution diffusion process toward the estimated distributions of condition fulfillment. We evaluate the performance of `CDGraph` on real social networks by measuring the dual-conditional validity and the discrepancy in various social network metrics. The contributions include:

- We derive the first kind of dual-conditional graph diffusion model, `CDGraph`, for synthesizing social graphs.
- We explore a novel feature of *co-evolution dependency* incorporating *social homophily* and *social contagion* to capture the dependency between specified conditions so as to preserve the structural information between nodes satisfying specified conditions.
- We introduce a novel loss function of the *dual-condition classifier* that guides the denoising process of `CDGraph` to jointly optimize the discrepancy in diffusion process and condition fulfillment.
- Through evaluations on four real-world social networks, we demonstrate that `CDGraph` outperforms the baselines in generating social graphs that satisfy the specified dual conditions and maintain social network properties.

2 RELATED WORK

Graph Generation. The current research on graph generation techniques can be divided into statistic-based and deep generative model based ones. The existing statistic-based graph generation is mainly based on structural information such as network statistics (Schweimer et al., 2022), correlation (Erling et al., 2015), community structure (Luo et al., 2020), and node degree (Wang et al., 2021), etc. There are several social graph generators for various purposes, e.g., frequent patterns (Shuai et al., 2013) and similarity across social network providers (Shuai et al., 2018). For deep generative model-based ones, they are based on auto-regression (Liao et al., 2019; Shi et al., 2020), variational autoencoder (Guo et al., 2021; Samanta et al., 2020), and GAN (Martinkus et al., 2022), etc. Especially, SPECTRE (Martinkus et al., 2022) is a GAN-based conditional generative model conditioning on graph Laplacian eigenvectors. However, both statistical and deep generative methods do not consider dual conditions, since they mainly simply focus on structural conditions or dependencies between consecutive steps, instead of the connectivity between users satisfying specified conditions (i.e., linking them according to the original structure), while avoiding to generate excessive irrelevant users for meaningful downstream analysis such as estimating social influence of a user to a specified population.

Diffusion Models. The current research direction on diffusion models mainly focuses on the application to multimedia, such as computer vision, text-image processing, and audio processing (Saharia et al., 2022; Gu et al., 2022; Hoogeboom et al., 2021; Savinov et al., 2022). Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020; Dhariwal & Nichol, 2021) has performed significantly better than generative adversarial networks in image synthesis. Recently, diffusion models have been also applied to generate graph data (e.g., molecular graph generation) (Niu et al., 2020; Huang et al., 2022; Vignac et al., 2023; Chen et al., 2023) thanks to the flexible modeling architecture and tractable probabilistic distribution compared with the aforementioned deep generative model architectures. In particular, DiGress (Vignac et al., 2023) synthesizes molecular graphs with the discrete denoising probabilistic model building on a discrete space. It exploits regression guidance to lead the denoising process to generate graphs that meet the condition property. EDGE (Chen et al., 2023) is a discrete diffusion model exploiting graph sparsity to generate graphs conditioning on the change of node degree. However, the above existing studies only deal with a single condition on nodes or edges in graphs, failing to capture the dependencies of conditions on graphs.

3 CDGRAPH

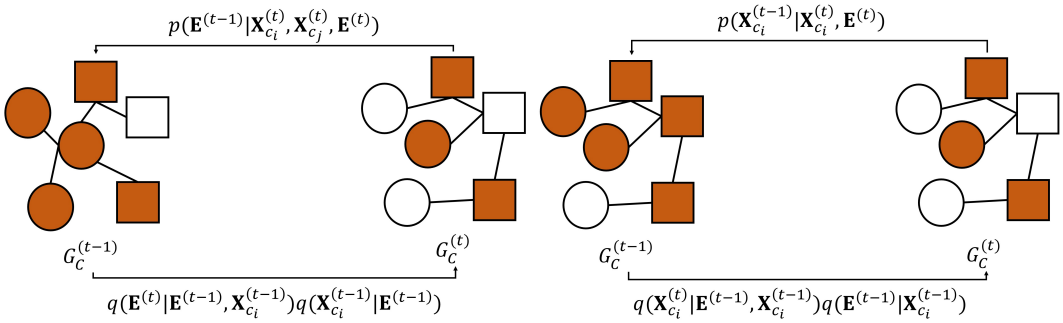
In this section, we begin by introducing the dual conditional graph generation problem. Subsequently, we revisit the concepts of the discrete diffusion model with a single condition of DiGress (Vignac et al., 2023) as a preliminary to our approach. Finally, we propose `CDGraph`, featuring on the novel loss, the co-evolution dependency incorporating social contagion and social homophily, and the dual-condition classifier guidance.

We first provide the technical intuition behind `CDGraph`. To guide the diffusion process with dual conditions, an intuitive approach is to extend the conditional DiGress (Vignac et al., 2023) by adding the guidance of the second condition. However, the dependencies across conditions are not explicitly captured and fall short in guiding the graph generation. Hence, we introduce the aforementioned dependencies to jointly optimize the structural similarity and condition satisfaction.

3.1 FORMAL PROBLEM DEFINITION

Here, we formulate the problem of *Dual Conditional Graph*. To perform the graph generation with the guidance of two conditions, we first define the *Condition Indication Graph* as follows. Figures 1(b) and 1(c) illustrate the two examples of conditional indication graphs.

Definition 1 (Condition Indication Graph)**.** *Let C denote the condition set. We define the condition indication graph of C by $G_C = (\{\mathbf{X}_c\}_{c \in C}, \mathbf{E})$, where $\mathbf{X}_c$ indicate nodes' satisfaction of condition c, and $\mathbf{E}_c$ indicates the existence of edges between nodes. Specifically, $\mathbf{x}_{n,c} \in \{0,1\}^2$ in $\mathbf{X}_c$ is a one-hot encoding vector representing whether a node v_n in G_C satisfies condition c; $e_{n,m} = (v_n, v_m)$ in $\mathbf{E}$ is a one-hot encoding vector representing whether an edge between v_n and v_m in G_C satisfies condition c.*



(a) Social Homophily (b) Social Contagion

Figure 2: Diffusion process in CDGraph with notions of social homophily and social contagion.

Definition 2 (Dual Conditional Graph Generation)**.** *Given the condition set $C = \{c_1, c_2\}$ and the condition indication graph $G_C = (\{\mathbf{X}_{c_1}, \mathbf{X}_{c_2}\}, \mathbf{E})$, the problem is to generate social graphs, such that i) the structural information of the generated graphs is similar to G_C, and ii) the majority of nodes in the generated graphs meet the conditions c_1 and c_2.*

3.2 Discrete Diffusion Models for Graph Generation

3.2.1 Preliminary: Discretizing Diffusion Profess

We revisit DiGress (Vignac et al., 2023), which is a discrete diffusion model with a single condition, i.e., $C = \{c\}$ and $G_C = (\{\mathbf{X}_c\}, \mathbf{E})$. Typically, DiGress consists of two components: forward noising process and reverse denoising process. For $t \geq 1$, the forward noising process of DiGress is defined by $q(G^{(t)}|G^{(t-1)})$ and $q(G^{(T)}|G^{(0)}) = \prod_{t=1}^{T} q(G^{(t)}|G^{(t-1)})$.

For the reverse denoising process, given $G^{(t)}$, DiGress predicts the clean graph $G^{(0)}$ by a denoising neural network ϕ_θ (parameterized by θ) and obtains the reverse denoising process p_θ as follows:

$$\begin{aligned} p_\theta(G^{(t-1)}|G^{(t)}) &= q(G^{(t-1)}|G^{(t)}, G^{(0)})p_\theta(G^{(0)}|G^{(t)}); \\ q(G^{(t-1)}|G^{(t)}) &\propto q(G^{(t)}|G^{(t-1)})q(G^{(t-1)}|G^{(0)}), \end{aligned} \tag{1}$$

in which $q(G^{(t-1)}|G^{(t)})$ can be approximated by the noising process. To enable single conditional graph generation, DiGress guides the reverse denoising process by a machine learning model f (i.e., a regression model) to push the predicted distribution toward graphs fulfilling the condition c. f is trained to predict the condition c of the input graph G from the noised version $G^{(t)}$ such that $c \approx \hat{c} = f(G^{(t)})$. The reverse denoising process guided by a single condition is presented as follows:

$$q(G^{(t-1)}|G^{(t)}, c) \propto q(c|G^{(t-1)})q(G^{(t-1)}|G^{(t)}), \tag{2}$$

where the first term is approximated by the learned distribution of the regression model, and the second term is approximated by the unconditional diffusion model.

3.3 Dual Conditional Graph Synthesizing

Here, we present the overall learning framework of CDGraph and its novel features for integrating dual conditions. Specifically, to capture the dependency between two exogenous conditions, CDGraph introduces *co-evolution dependency* to model the co-evolving diffusion process of dual conditions in the reverse denoising process with the notions of *social homophily* and *social contagion* so that the dependencies between node conditions and edge connections can be captured in the diffusion process. Then, to further fulfill dual conditions, we design a *dual condition classifier* to guide the co-evolution diffusion process, modulating the estimated distribution to align with graphs satisfying specified conditions by the classifier loss.

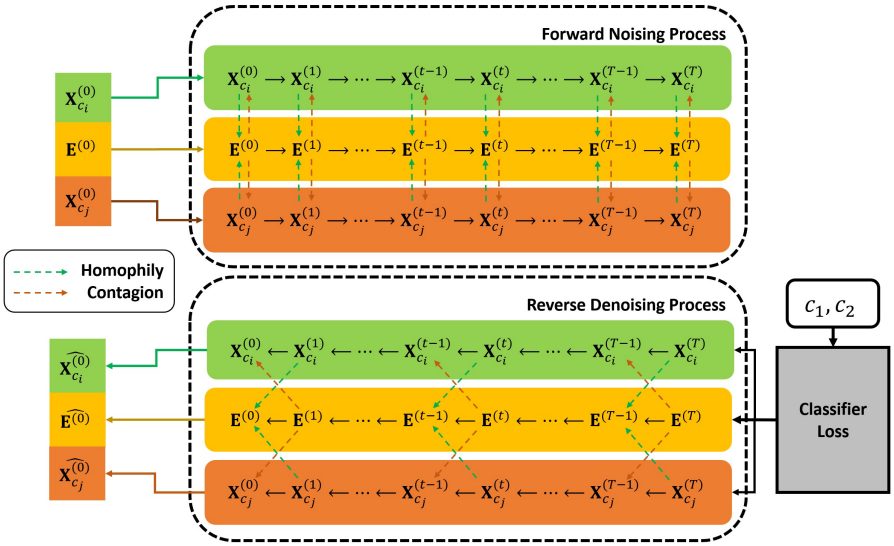


Figure 3: Workflow of CDGraph.

Specifically, CDGraph comprises the forward noising process and the reverse denoising process. Given $C = \{c_1, c_2\}$ and the condition indication graph $G_C = (\{\mathbf{X}_{c_1}, \mathbf{X}_{c_2}\}, \mathbf{E})$, we model the forward noising processes for each of the specified conditions $c_i, \forall i \in \{1, 2\}$ as follows:

$$q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)}) = \mathbf{X}_{c_i}^{(t-1)}\mathbf{Q}_{X_{c_i}}^{(t)}; q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)}) = \mathbf{E}^{(t-1)}\mathbf{Q}_{E}^{(t)},$$

where $\mathbf{Q}_{X_{c_i}}^{(t)}$ and $\mathbf{Q}_{E}^{(t)}$ are transition matrices for $\mathbf{X}_{c_i}$ and $\mathbf{E}$, respectively. Then we can show that $q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)})$ and $q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)})$ obey Bernoulli distributions as follows:

$$q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)}) = \mathcal{B}(\mathbf{X}_{c_i}^{(t)}; (1-\beta_t)\mathbf{X}_{c_i}^{(t-1)} + \beta_t \mathbf{1}/2); \tag{3}$$

$$q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)}) = \mathcal{B}(\mathbf{E}^{(t)}; (1-\beta_t)\mathbf{E}^{(t-1)} + \beta_t \mathbf{1}/2). \tag{4}$$

The detailed derivations regarding $\mathbf{Q}_{X_{c_i}}^{(t)}$ and $\mathbf{Q}_{E}^{(t)}$ are provided in Appendix A.

The notions of *social homophily* and *social contagion* in the diffusion process in CDGraph are illustrated in Figure 2. In Fig. 2(a), as denoising from $G_C^{(t)}$ to $G_C^{(t-1)}$, the nodes with similar attributes (orange nodes) in $G_C^{(t-1)}$ tend to have links between them. In Fig. 2(b), as denoising from $G_C^{(t)}$ to $G_C^{(t-1)}$, the edges incident to orange-colored nodes tend to cause the other node have the same attribute.

By considering nodes with two conditions and the dependency between nodes and edges, the overall forward processes of $\mathbf{X}_{c_i}$ and $\mathbf{E}$ are formulated as follows:

$$q(\mathbf{X}_{c_i}^{(0:T)}) = \prod_{t=1}^{T} q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)}, \mathbf{E}^{(t-1)}) q(\mathbf{E}^{(t-1)}|\mathbf{X}_{c_i}^{(t-1)}),$$

$$q(\mathbf{E}^{(0:T)}) = \prod_{t=1}^{T} q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)}, \mathbf{X}_{c_i}^{(t-1)}) q(\mathbf{X}_{c_i}^{(t-1)}|\mathbf{E}^{(t-1)}). \tag{5}$$

The concept of the above process is illustrated in the upper half of Figure 3, in which the green dashed lines represent the *social homophily*, whereas the brown dashed lines represent the *social contagion*. The former facilitates the connections between nodes with similar conditions; the latter makes adjacent nodes to have similar conditions. The detailed forward transition distribution with dependency in the above processes is defined and analyzed in Appendix A.

3.3.1 CO-EVOLUTION DEPENDENCY

Different from DiGress, `CDGraph` exploits *co-evolution dependency* to model the intricate dependency across conditions of the nodes and connections between them in the denoising process, which is detailed in Appendix A. This emphasis on capturing the relationship between $\mathbf{X}_{c_i}^{(t)}, i \in \{1, 2\}$ and $\mathbf{E}^{(t)}$ are crucial for the precise reconstruction of the input graphs. To capture the dependencies between the connection between nodes and condition satisfaction of the nodes in a social graph, we build a denoising model incorporating two phenomena in social networks: *social homophily* and *social contagion*. The concept of the above denoising process is illustrated in the lower half of Figure 3.

Assuming that $\mathbf{X}_{c_i}^{(t-1)}$, $\mathbf{X}_{c_j}^{(t-1)}$ and $\mathbf{E}^{(t-1)}$ are conditionally independent given $\mathbf{X}_{c_i}^{(t)}$, $\mathbf{X}_{c_j}^{(t)}$ and $\mathbf{E}^{(t)}$, the reverse denoising process can be further decomposed as follows:

$$\begin{aligned} &p_\theta(\mathbf{X}_{c_i}^{(t-1)}, \mathbf{X}_{c_j}^{(t-1)}, \mathbf{E}^{(t-1)} | \mathbf{X}_{c_i}^{(t)}, \mathbf{X}_{c_j}^{(t)}, \mathbf{E}^{(t)}) \\ &= p_\theta(\mathbf{X}_{c_i}^{(t-1)} | \mathbf{X}_{c_i}^{(t)}, \mathbf{E}^{(t)}) p_\theta(\mathbf{X}_{c_j}^{(t-1)} | \mathbf{X}_{c_j}^{(t)}, \mathbf{E}^{(t)}) p_\theta(\mathbf{E}^{(t-1)} | \mathbf{X}_{c_i}^{(t)}, \mathbf{E}^{(t)}, \mathbf{X}_{c_j}^{(t)}), \end{aligned} \tag{6}$$

in which the first two terms represent *social contagion* that can be used to denoise node conditions from given edges, and the third term represents *social homophily* that can be exploited to denoise connections between nodes from given conditions of nodes. Note that $\mathbf{X}_{c_i}^{(t-1)}$ is independent on $\mathbf{X}_{c_j}^{(t)}$ (and $\mathbf{X}_{c_j}^{(t-1)}$ is independent on $\mathbf{X}_{c_i}^{(t)}$) for distinct c_i and c_j.

Social Homophily-based Co-evolution. In this paragraph, we discuss how to guide the diffusion process with social homophily, which states that nodes with similar conditions tend to have edges between them. Accordingly, we consider the following denoising process:

$$\begin{aligned} p_\theta(\mathbf{E}^{(0:T)}) &= p_\theta(\mathbf{X}_{c_i}^{(T)}) \prod_{t=1}^{T} p_\theta(\mathbf{E}^{(t-1)} | \mathbf{E}^{(t)}, \mathbf{X}_{c_i}^{(t)}); \\ p_\theta(\mathbf{E}^{(t-1)} | \mathbf{E}^{(t)}, \mathbf{X}_{c_i}^{(t)}) &= \mathcal{B}(\mathbf{p}_\theta^{(homo)}), \end{aligned} \tag{7}$$

where

$$\mathbf{p}_\theta^{(homo)} = \sum_{\hat{\mathbf{E}}^{(0)} \in \{0,1\}} q(\mathbf{E}^{(t-1)} | \mathbf{E}^{(t)}, \hat{\mathbf{E}}^{(0)}) \hat{p}_e(\hat{\mathbf{E}}^{(0)} | \mathbf{E}^{(t)}, \mathbf{X}_{c_i}^{(t)}),$$

and $\hat{p}_e$ is the distribution learned to predict $\mathbf{E}^{(0)}$ from $\mathbf{E}^{(t)}$ conditioned on $\mathbf{X}_{c_i}^{(t)}, \mathbf{X}_{c_i}^{(t)}$ by denoising network ϕ_θ.

The loss function of social homophily-based co-evolving diffusion can be derived as follows:

$$\begin{aligned} \mathcal{L}_{homo} = &\sum_{t=2}^{T-1} D_{KL}[q(\mathbf{E}^{(t-1)} | \mathbf{E}^{(t)}, \mathbf{E}^{(0)}) \| p_\theta(\mathbf{E}^{(t-1)} | \mathbf{E}^{(t)}, \mathbf{X}_{c_i}^{(t)}, \mathbf{X}_{c_j}^{(t)})] \\ &+ D_{KL}[q(\mathbf{E}^{(T)} | \mathbf{E}^{(0)}) \| p(\mathbf{E}^{(T)})] - \log p_\theta(\mathbf{E}^{(0)} | \mathbf{E}^{(1)}, \mathbf{X}_{c_i}^{(1)}, \mathbf{X}_{c_j}^{(1)}), \end{aligned} \tag{8}$$

where the first term is the loss for diffusion process; the second term is the loss for prior distribution; the third term is the loss for reconstruction.

Social Contagion-based Co-evolution. In this paragraph, we discuss how to guide the diffusion process with social contagion, which states that nodes connected with edges tend to have similar conditions. Accordingly, we consider the following denoising process:

$$\begin{aligned} p_\theta(\mathbf{X}_{c_i}^{(0:T)}) &= p_\theta(\mathbf{E}^{(T)}) \prod_{t=1}^{T} p_\theta(\mathbf{X}_{c_i}^{(t-1)} | \mathbf{X}_{c_i}^{(t)}, \mathbf{E}^{(t)}); \\ p_\theta(\mathbf{X}_{c_i}^{(t-1)} | \mathbf{X}_{c_i}^{(t)}, \mathbf{E}^{(t)}) &= \mathcal{B}(\mathbf{p}_\theta^{(cont)}), \end{aligned} \tag{9}$$

where

$$\mathbf{p}_\theta^{(cont)} = \sum_{\hat{\mathbf{X}}_{c_i}} q(\mathbf{X}_{c_i}^{(t-1)}|\mathbf{X}_{c_i}^{(t)}, \hat{\mathbf{X}}_{c_i}^{(0)})\hat{p}_{c_i}(\hat{\mathbf{X}}_{c_i}^{(0)}|\mathbf{X}_{c_i}^{(t)}, \mathbf{E}^{(t)}),$$

and $\hat{p}_{c_i}$ is the distribution learned to predict $\mathbf{X}_{c_i}^{(0)}$ from $\mathbf{X}_{c_i}^{(t)}$ conditioned on $\mathbf{E}^{(t)}$ by denoising network ϕ_θ.

And the loss function for social contagion-based co-evolution can be derived as follows:

$$\begin{aligned}\mathcal{L}_{cont} = &\sum_{c_i \in C}\sum_{t=2}^{T-1} D_{KL}[q(\mathbf{X}_{c_i}^{(t-1)}|\mathbf{X}_{c_i}^{(t)}, \mathbf{X}_{c_i}^{(0)})\|p_\theta(\mathbf{X}_{c_i}^{(t-1)}|\mathbf{X}_{c_i}^{(t)}, \mathbf{E}^{(t)})] \\ &+ D_{KL}[q(\mathbf{X}_{c_i}^{(T)}|\mathbf{X}_{c_i}^{(0)})\|p(\mathbf{X}_{c_i}^{(T)})] - \log p_\theta(\mathbf{X}_{c_i}^{(0)}|\mathbf{X}_{c_i}^{(1)}, \mathbf{E}^{(1)}). \end{aligned} \tag{10}$$

The overall loss function of the co-evolution diffusion process is $\mathcal{L} = \mathcal{L}_{homo} + \mathcal{L}_{cont}$, which jointly optimizes the discrepancy in diffusion process and graph reconstruction in order to synthesize graphs with properties of social homophily and social contagion. The pseudocode of training and sampling is depicted in Appendix B.

3.3.2 Dual-condition Classifier

Afterward, `CDGraph` leverages *dual conditional classifier* to enable joint guidance for $p_\theta(G_C^{(t-1)}|G_C^{(t)})$ to fulfill dual conditions, instead of single conditional guidance in DiGress. To guide the diffusion process with two specified conditions jointly, we exploit the concept of conditional guidance and model the guidance distribution for the specified conditions c_i and c_j ($i \neq j$) with a classifier such that the generated graphs are classified according to whether a majority of the nodes satisfy both of the specified conditions or not. Note that the diffusion process satisfies the Markovian property: $q(G_C^{(t-1)}|G_C^{(t)}, c_i) = q(G_C^{(t-1)}|G_C^{(t)}), \forall i$.

Thus, from the results derived in the single-conditional denoising process, the co-evolution reverse denoising process with dual-condition classifier can be derived as follows:

$$\begin{aligned} q(G_C^{(t-1)}|G_C^{(t)}, c_i, c_j) &= \frac{q(c_i|G_C^{(t-1)}, G_C^{(t)}, c_j)q(G_C^{(t-1)}, G_C^{(t)}, c_j)}{q(c_i|G_C^{(t)}, c_j)q(c_j|G_C^{(t)})q(G_C^{(t)})} \\ &= \frac{q(c_i|G_C^{(t-1)}, G_C^{(t)}, c_j)q(G_C^{(t-1)}|G_C^{(t)}, c_j)}{q(c_i|G_C^{(t)}, c_j)} \propto q(c_i|G_C^{(t-1)}, c_j)q(c_j|G_C^{(t-1)})q(G_C^{(t-1)}|G_C^{(t)}), \end{aligned}$$

where the first two terms enable the hierarchical guidance of the conditions to guide the denoised graph $G_C^{(t-1)}$ satisfying one condition c_j (i.e., more than half of the nodes in $G_C^{(t-1)}$ satisfy condition c_j), and then satisfying the other condition c_i (i.e., more than half of the nodes in $G_C^{(t-1)}$ satisfy condition c_i) given that $G_C^{(t-1)}$ satisfies condition c_j. The third term $q(G_C^{(t-1)}|G_C^{(t)})$ is approximated by the aforementioned co-evolution denoising process with *social homophily* and *social contagion*. Note that, compared to DiGress and other previous studies, the proposed guidance model has more capability to guide the denoising process such that the denoised graphs meet both of the specified conditions if the specified conditions have a negative correlation. The pseudocode of conditional sampling is depicted in Appendix B.

4 Empirical Evaluation

4.1 Setup

Datasets. We conduct the experiments on the following datasets shown in Table 1. We sample ego networks with at maximum 100 nodes for each dataset (Facebook: 3548 graphs; Twitter: 75940 graphs; Flickr: 6450 graphs; BlogCatalog: 4219 graphs).

Baselines. We compare `CDGraph` with the following baselines. (1) SPECTRE (Martinkus et al., 2022): A GAN-based graph generator modeling dominant components and generating graphs conditioning on graph Laplacian eigenvectors. (2) GSM (Niu et al., 2020): A permutation invariant

Table 1: Dataset descriptions

	Facebook	Twitter	Flickr	BlogCatalog
#Node	4.03K	81.3K	7.57K	5.19K
#Edges	88.23K	216.83K	479.47K	343.48K

Table 2: Feasibility when the dual conditions are weakly positively correlated.

Dataset	Facebook					BlogCatalog				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.448	0.243	1.795	1.119	0.313	0.3	0.026	6.081	5.494	0.098
GSM	0.272	**0.100**	0.551	0.392	0.031	0.29	**0.013**	3.486	3.441	0.063
EDGE	0.262	0.736	1.983	0.815	0.028	0.292	0.414	0.510	0.256	**0.014**
DiGress	0.375	0.111	0.852	0.457	0.143	0.3125	0.164	0.413	0.699	0.266
CDGraph	**1**	0.149	**0.186**	**0.002**	**0.022**	**1**	0.020	**0.259**	**0.219**	0.031
Dataset	Twitter					Flickr				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.239	0.163	1.061	0.131	0.359	0.323	0.053	2.115	2.275	1.006
GSM	0.309	0.205	0.417	**0.034**	0.011	0.269	**0.005**	1.055	1.039	**0.014**
EDGE	0.291	0.101	0.034	0.302	0.002	0.286	0.439	0.809	0.695	**0.014**
DiGress	0.5	**0.029**	0.301	0.184	0.174	0.375	0.090	0.233	0.139	0.286
CDGraph	**1**	0.045	**0.001**	0.409	**0.016**	**1**	0.008	**0.056**	**0.068**	0.068

score-based graph generator modeling the gradient of the data distribution at the input graph with a multi-channel GNN architecture. (3) EDGE (Chen et al., 2023): A discrete diffusion model that explicitly generates graphs conditioned on changes in node degree. (4) DiGress (Vignac et al., 2023): A discrete diffusion model that generates graphs by a categorical distribution of node/edge types (also with a regression guidance condition).[1]

Evaluation Metrics. We conduct comparative studies by evaluating the following metrics. (1) Validity (the higher, the better): It evaluates the proportion of generated graphs with a majority of nodes meeting both specified conditions. (2) Relative error ratios (the lower, the better): It evaluates the relative differences of the average numbers of nodes and edges, as well as the average density, between the input and generated graphs. (3) MMD (maximum mean discrepancy; the lower, the better): It evaluates the discrepancy in the distributions of clustering coefficients between the input and generated graphs.[2]

4.2 Feasibility on social graph generation

In this section, we examine the dual conditions in two cases: those that are weakly positively correlated and those that are weakly negatively correlated. Additional cases are presented in Appendix C due to space constraints. Note that since the conditions in Twitter and Flickr are all positively correlated, only the results from Facebook and BlogCatalog are reported for the case with weakly negatively correlated conditions. Tables 2 and 3 demonstrate the feasibility of `CDGraph` and baselines on social graph generation in terms of validity, relative error ratios, and MMD for the cases with weakly positively correlated and weakly negatively correlated, respectively, conditions.

Validity. Compared with the baselines, `CDGraph` is demonstrated to be able to generate graphs with the majority of nodes satisfying both of the specified conditions since it exploits the guidance of the dual condition classifier. Among the baselines, DiGress usually achieves the highest validity since it is aware of dual conditions, whereas SPECTRE, GSM, and EDGE do not account for them. Nevertheless, DiGress neglects the dependency between dual conditions and only achieves half of the validity achieved by `CDGraph`.

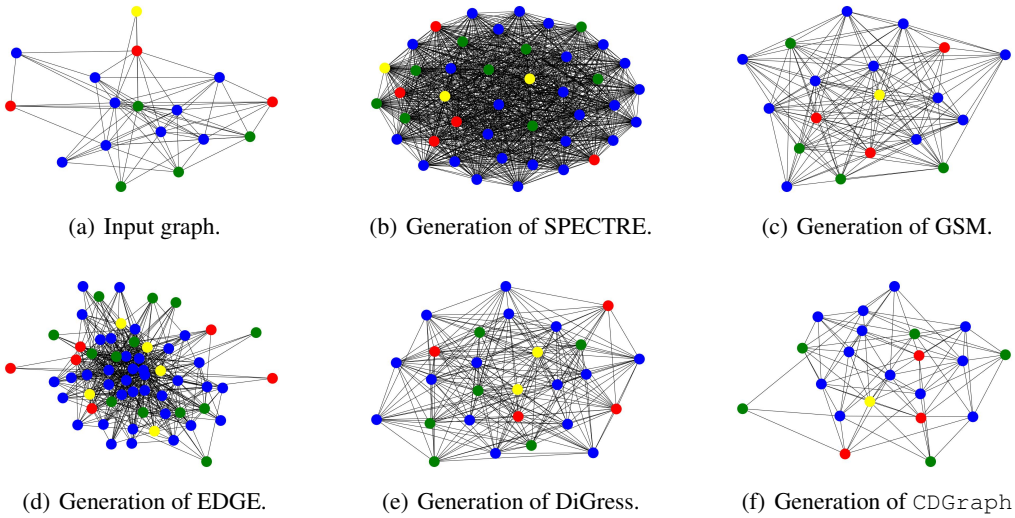
Relative error ratios of #nodes, #edges, and density, and MMD of clustering coefficients. `CDGraph` has superior performance in the relative error ratios of the edge number and density on Facebook and BlogCatalog for both cases of positive and negative correlations of specified condi-

[1] Since DiGress only considers a single condition, we extend it to handle the dual conditions by multiplying the guidance models of the respective conditions.

[2] Since we are primarily concerned with users who meet the specified conditions, we consider the subgraphs induced by those nodes that satisfy both specified conditions as the input graphs when evaluating the relative error ratios and MMD.

Table 3: Feasibility when the dual conditions are weakly negatively correlated.

Dataset	Facebook					BlogCatalog				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.241	0.241	0.323	0.254	0.307	0.2	0.066	4.271	4.260	0.091
GSM	0.207	0.176	0.837	0.506	0.032	0.244	0.011	3.462	3.430	0.074
EDGE	0.256	0.805	2.135	0.852	0.03	0.276	0.375	0.513	0.299	**0.013**
DiGress	0.375	0.341	0.545	0.093	0.361	0.438	**0.005**	1.066	1.028	0.263
CDGraph	**0.94**	**0.102**	**0.261**	**0.060**	**0.016**	**1**	0.020	**0.259**	**0.219**	0.031



(a) Input graph. (b) Generation of SPECTRE. (c) Generation of GSM.

(d) Generation of EDGE. (e) Generation of DiGress. (f) Generation of CDGraph

Figure 4: Visualization results of Twitter.

tions. Regarding the MMD of clustering coefficients, CDGraph demonstrates the best performance on Facebook and Twitter and also ranks second on Flickr and BlogCatalog. As the clustering coefficient is a fundamental property of social graphs, the results highlight that CDGraph surpasses the baselines in generating social graphs.

Visualization To better understand the ability of social graph generation, we further present the visualization results. Figure 4 visualizes the input and generated graphs for Twitter. Nodes in blue indicate that both specified conditions are met, while nodes in yellow and green represent those satisfying only one condition. Conversely, nodes in red do not meet any of the conditions. Obviously, the generation results of CDGraph are the closest to the input graph, thanks to the denoising process based on social homophily and social contagion. Due to the space constraint, additional visualization results are presented in Appendix C.

5 Conclusion

In this paper, we make the first attempt to develop a conditional diffusion model for social networks, CDGraph, which aims to synthesize social graphs satisfying two specified conditions. In contrast to previous studies on deep generative (diffusion) model-based graph generation, CDGraph introduces a novel feature of *co-evolution dependency*. This feature integrates the concepts of *social homophily* and *social contagion*, allowing CDGraph to capture the interdependencies between specified conditions. Moreover, CDGraph exploits the *dual-condition classifier* in the denoising process, ensuring that discrepancies in the diffusion process and structure reconstruction are jointly optimized. The experimental results manifest that CDGraph achieves lower discrepancies in various network statistics under various correlations among specified conditions compared with the baselines.

REFERENCES

Angela Bonifati, Irena Holubová, Arnau Prat-Pérez, and Sherif Sakr. Graph generators: State of the art and open challenges. 53(2), apr 2020. ISSN 0360-0300. doi: 10.1145/3379445. URL https://doi.org/10.1145/3379445.

Xiaohui Chen, Jiaxing He, Xu Han, and Li-Ping Liu. Efficient and degree-guided graph generation via discrete diffusion modeling. In *International Conference on Machine Learning*, 2023.

Vijil Chenthamarakshan, Payel Das, Samuel Hoffman, Hendrik Strobelt, Inkit Padhi, Kar Wai Lim, Benjamin Hoover, Matteo Manica, Jannis Born, Teodoro Laino, and Aleksandra Mojsilovic. Cogmol: Target-specific and selective drug design for covid-19 using deep generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4320–4332. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/2d16ad1968844a4300e9a490588ff9f8-Paper.pdf.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021.

Orri Erling, Alex Averbuch, Josep Larriba-Pey, Hassan Chafi, Andrey Gubichev, Arnau Prat, Minh-Duc Pham, and Peter Boncz. The ldbc social network benchmark: Interactive workload. SIGMOD '15, pp. 619–630, New York, NY, USA, 2015. Association for Computing Machinery. doi: 10.1145/2723372.2742786.

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 (CVPR)*, pp. 10696–10706, June 2022.

Xiaojie Guo, Yuanqi Du, and Liang Zhao. Deep generative models for spatial networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 505–515, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467394. URL https://doi.org/10.1145/3447548.3467394.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. Conditional diffusion based on discrete graph structures for molecular graph generation. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Wenjian Luo, Binyao Duan, Hao Jiang, and Li Ni. Time-evolving social network generator based on modularity: Tesng-m. *IEEE Transactions on Computational Social Systems*, 7(3):610–620, 2020. doi: 10.1109/TCSS.2020.2979806.

Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. SPECTRE: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15159–15179. PMLR, 17–23 Jul 2022.

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494. Curran Associates, Inc., 2022.

Bidisha Samanta, Abir De, Gourhari Jana, Vicenç Gomez, Pratim Kumar Chattaraj, Niloy Ganguly, and Manuel Gomez-Rodriguez. Nevae: A deep generative model for molecular graphs. 21(1), jan 2020. ISSN 1532-4435.

Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*, 2022.

Christoph Schweimer, Christine Gfrerer, Florian Lugstein, David Pape, Jan A. Velimsky, Robert Elsässer, and Bernhard C. Geiger. Generating simple directed social network graphs for information spreading. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pp. 1475–1485, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512194. URL https://doi.org/10.1145/3485447.3512194.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2020.

Hong-Han Shuai, De-Nian Yang, Philip S. Yu, Chih-Ya Shen, and Ming-Syan Chen. On pattern preserving graph generation. In *2013 IEEE 13th International Conference on Data Mining*, pp. 677–686, 2013. doi: 10.1109/ICDM.2013.14.

Hong-Han Shuai, De-Nian Yang, Chih-Ya Shen, Philip S. Yu, and Ming-Syan Chen. Qmsampler: Joint sampling of multiple networks with quality guarantee. *IEEE Transactions on Big Data*, 4 (1):90–104, 2018. doi: 10.1109/TBDATA.2017.2715847.

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.

Chaokun Wang, Binbin Wang, Bingyang Huang, Shaoxu Song, and Zai Li. Fastsgg: Efficient social graph generation using a degree distribution generation model. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 564–575, 2021. doi: 10.1109/ICDE51399.2021.00055.

A DETAILED DERIVATIONS OF THE FORWARD PROCESSES

As mentioned in Sec. 3, each step of the forward noising processes of $\mathbf{X}_{c_i}$ (for each of the specified conditions $c_i, i \in \{1, 2\}$) and $\mathbf{E}$ can be derived as follows:

$$q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)}) = \mathbf{X}_{c_i}^{(t-1)}\mathbf{Q}_{X_{c_i}}^{(t)}; q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)}) = \mathbf{E}^{(t-1)}\mathbf{Q}_{E_{c_i}}^{(t)},$$

where the transition matrices $\mathbf{Q}_{X_{c_i}}^{(t)}, \mathbf{Q}_{E_{c_i}}^{(t)} \in \mathbb{R}^{2\times 2}$ are defined with a parameter $\beta_t \in (0, 1)$ as follows:

$$\mathbf{Q}_{X_{c_i}}^{(t)} = \begin{bmatrix} 1-\beta_t & \beta_t \\ \beta_t & 1-\beta_t \end{bmatrix}; \mathbf{Q}_{E_{c_i}}^{(t)} = \begin{bmatrix} 1-\beta_t & \beta_t \\ \beta_t & 1-\beta_t \end{bmatrix}.$$

Note that it is equivalent to express $q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)})$ and $q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)})$ in the form of the Bernoulli distribution $\mathcal{B}$ as follows:

$$\begin{aligned} q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)}) &= \mathcal{B}(\mathbf{X}_{c_i}^{(t)}; (1-\beta_t)\mathbf{X}_{c_i}^{(t-1)} + \beta_t \mathbf{1}/2); \\ q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)}) &= \mathcal{B}(\mathbf{E}^{(t)}; (1-\beta_t)\mathbf{E}^{(t-1)} + \beta_t \mathbf{1}/2). \end{aligned} \tag{11}$$

Recall that the overall forward processes of $\mathbf{X}_{c_i}$ and $\mathbf{E}$ are formulated as follows:

$$\begin{aligned} q(\mathbf{X}_{c_i}^{(0:T)}) &= \prod_{t=1}^{T} q(\mathbf{X}_{c_i}^{(t)}|\mathbf{X}_{c_i}^{(t-1)}, \mathbf{E}^{(t-1)}) q(\mathbf{E}^{(t-1)}|\mathbf{X}_{c_i}^{(t-1)}), \\ q(\mathbf{E}^{(0:T)}) &= \prod_{t=1}^{T} q(\mathbf{E}^{(t)}|\mathbf{E}^{(t-1)}, \mathbf{X}_{c_i}^{(t-1)}) q(\mathbf{X}_{c_i}^{(t-1)}|\mathbf{E}^{(t-1)}), \end{aligned} \tag{12}$$

where the dependency between $\mathbf{X}_{c_i}^{(t-1)}$ and $\mathbf{E}^{(t-1)}$ can be determined by the properties of social homophily and social contagion as follows.

For social contagion, given the existence of an edge $e_{m,n}$ (i.e., $\mathbf{e}_{m,n}^{(t-1)} = 1$), we define the probability whether node x_m satisfies condition c_i (i.e., $\mathbf{x}_{c_i,m}^{(t-1)} = 1$) or not as follows.

$$q(\mathbf{x}_{c_i,m}^{(t-1)}|\mathbf{e}_{m,n}^{(t-1)} = 1, \mathbf{x}_{c_i,n}^{(t-1)}) = \begin{cases} p, \mathbf{x}_{c_i,m}^{(t-1)} = \mathbf{x}_{c_i,n}^{(t-1)}, \\ 1-p, \mathbf{x}_{c_i,m}^{(t-1)} \neq \mathbf{x}_{c_i,n}^{(t-1)}. \end{cases} \tag{13}$$

For social homophily, we define the probability of the existence of an edge $\mathbf{e}_{m,n}$ from given node condition $\mathbf{x}_{c_i,m}^{(t-1)}$ by marginalization with respect to another node condition $\mathbf{x}_{c_i,n}^{(t-1)}$ as follows:

$$q(\mathbf{e}_{m,n}^{(t-1)} = 1|\mathbf{x}_{c_i,m}^{(t-1)}) = \sum_{\mathbf{x}_{c_i,n}^{(t-1)} \in \{0,1\}} q(\mathbf{e}_{m,n}^{(t-1)} = 1|\mathbf{x}_{c_i,m}^{(t-1)}, \mathbf{x}_{c_i,n}^{(t-1)}) q(\mathbf{x}_{c_i,n}^{(t-1)}|\mathbf{x}_{c_i,m}^{(t-1)}), \tag{14}$$

where $q(\mathbf{x}_{c_i,n}^{(t-1)}|\mathbf{x}_{c_i,m}^{(t-1)})$ can be computed with the bivariate Bernoulli distribution of $\mathbf{x}_{c_i,n}^{(t-1)}$ and $\mathbf{x}_{c_i,m}^{(t-1)}$, and

$$q(\mathbf{e}_{m,n}^{(t-1)} = 1|\mathbf{x}_{c_i,m}^{(t-1)}, \mathbf{x}_{c_i,n}^{(t-1)}) = \begin{cases} 1, \mathbf{x}_{c_i,m}^{(t-1)} = \mathbf{x}_{c_i,n}^{(t-1)}, \\ 0, \mathbf{x}_{c_i,m}^{(t-1)} \neq \mathbf{x}_{c_i,n}^{(t-1)}. \end{cases}$$

B PSEUDOCODE OF THE PROPOSED ALGORITHMS

The training procedure of `CDGraph` is depicted in Algorithm 1.

The sampling procedure of `CDGraph` is depicted in Algorithm 2.

The pseudocode of the sampling of the diffusion process is shown in the Algorithm 3.

Algorithm 1 `CDGraph` Training

Require: A condition indication graph $G_C = (\{\mathbf{X}_{c_i}\}_{i=1}^{|C|}, \mathbf{E})$ with condition set $C = \{c_1, c_2\}$.
Sample $t \sim \{1, 2, \cdots, T\}$
Sample $G_C^{(t)} \sim (\{\mathbf{X}_{c_i}\overline{\mathbf{Q}}_{X_{c_i}}^{(t)}\}_{c_i \in C}, \mathbf{E}\overline{\mathbf{Q}}_E^{(t)})$
Extract structural and spectral features $z \longleftarrow f(G_C^t, t)$
$\hat{p}_{X_{c_i}}, \hat{p}_E \longleftarrow \phi_\theta(G_C^t, z)$
Optimize the cross-entropy loss $l_{CE}(\hat{p}_{X_{c_1}}, \mathbf{X}_{c_1}) + l_{CE}(\hat{p}_{X_{c_2}}, \mathbf{X}_{c_2}) + \lambda l_{CE}(\hat{p}_E, \mathbf{E})$

Algorithm 2 Sampling from `CDGraph`

Sample n nodes from training data distribution
Sample $G_C^T \sim (\{q(\mathbf{X}_{c_i}^{(T)})\}_{c_i \in C}, q(\mathbf{E}^{(T)}))$
for $t = T$ to 1 **do**
Extract structural and spectral features $z \longleftarrow f(G_C^t, t)$
$\hat{p}_{X_{c_1}}, \hat{p}_E \longleftarrow \phi_\theta(G_C^t, z)$
$p_\theta(x_{n,c_i}^{(t-1)} | x_{n,c_i}^{(t)}, e_{n,m}^{(t)}) \longleftarrow \sum_{x_{n,c_i}^{(0)} \in \{0,1\}} q(x_{n,c_i}^{(t-1)} | x_{n,c_i}^{(t)}, x_{n,c_i}^{(0)}) \hat{p}_{X_{c_i}}(x_{n,c_i}^{(0)} | x_{n,c_i}^{(t)}, e_{n,m}^{(t)})$ for $c_i \in C$.
$p_\theta(e_{n,m}^{(t-1)} | e_{n,m}^{(t)}, x_{n,C}^{(t)}, x_{m,C}^{(t)}) \longleftarrow \sum_{e_{n,m}^{(0)} \in \{0,1\}} q(e_{n,m}^{(t-1)} | e_{n,m}^{(t)}, e_{n,m}^{(0)}) \hat{p}_E(e_{n,m}^{(0)} | e_{n,m}^{(t)}, x_{n,C}^{(t)}, x_{m,C}^{(t)})$
Sample $\mathbf{X}_{c_i}^{(t-1)} \sim \prod_{n=1}^N \prod_{(n,m) \in E} p_\theta(x_{n,c_i}^{(t-1)} | x_{n,c_i}^{(t)}, e_{n,m}^{(t)})$
Sample $\mathbf{E}^{(t-1)} \sim \prod_{n=1}^N \prod_{(n,m) \in E} p_\theta(e_{n,m}^{(t-1)} | e_{n,m}^{(t)}, x_{n,C}^{(t)}, x_{m,C}^{(t)})$
end for
Ensure: $\hat{G}_C = (\{\hat{\mathbf{X}}_{c_i}\}_{c_i \in C}, \hat{\mathbf{E}})$

Algorithm 3 Conditional Sampling

Require: A condition indication graph $G_C = (\{\mathbf{X}_{c_i}\}_{i=1}^{|C|}, \mathbf{E})$ with condition set C.
Sample $G_C^T \sim (q(\mathbf{X}_{c_i}^{(T)}), q(\mathbf{E}^{(T)}))$
for $t = T$ to 1 **do**
$\hat{p}_{c_i}, \hat{p}_e \leftarrow \phi_\theta(G_C^{(t)})$
$\hat{c}_i \leftarrow g_{c_i}(G_C^{(t)})$
$\hat{c}_j \leftarrow g_{c_j}(G_C^{(t)})$ for $G_C^{(t)}$ with $\hat{c}_i = 1$.
$G_C^{(t-1)} \sim p_{c_j}(\hat{c_j} | G_C^{(t-1)}, c_i) p_{c_i}(\hat{c_i} | G_C^{(t-1)}) p_\theta(G_C^{(t-1)} | G_C^{(t)})$
end for
Ensure: $\hat{G}_C = (\{\hat{\mathbf{X}}_{c_i}\}_{c_i \in C}, \hat{\mathbf{E}})$

Dataset	Facebook					BlogCatalog				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.379	0.084	1.484	1.021	0.534	0.38	0.004	5.621	5.135	0.099
GSM	0.33	**0.083**	0.445	0.360	0.03	0.289	**0.002**	3.398	3.404	0.072
EDGE	0.307	0.919	2.528	0.934	0.029	0.282	0.420	0.562	0.323	**0.013**
DiGress	0.188	0.341	0.446	0.097	0.49	0.1875	0.063	0.779	0.892	0.342
CDGraph	**1**	0.149	**0.186**	**0.002**	**0.022**	**1**	0.020	**0.259**	**0.219**	0.031
Dataset	Twitter					Flickr				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.304	0.239	1.125	**0.001**	0.515	0.226	0.025	2.294	2.373	1.006
GSM	0.294	0.212	0.404	0.045	0.011	0.288	**0.007**	1.055	1.033	**0.013**
EDGE	0.299	0.107	0.042	0.304	**0.002**	0.265	0.427	0.799	0.681	0.014
DiGress	0.375	0.165	0.071	0.284	0.257	0.5	0.039	0.129	0.089	0.307
CDGraph	**1**	**0.045**	**0.001**	0.409	0.016	**1**	0.008	**0.056**	**0.068**	0.068

Table 4: Summarization of evaluation results ($0.125 < \rho < 0.25$)

Dataset	Facebook					BlogCatalog				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.276	0.292	2.341	1.190	0.217	0.26	0.018	5.800	5.304	0.096
GSM	0.298	**0.090**	0.433	0.371	0.029	0.242	0.024	3.562	3.472	0.078
EDGE	0.32	0.782	2.039	0.817	0.024	0.26	0.411	0.559	0.334	**0.012**
DiGress	0.5	0.470	0.664	0.286	0.244	0.313	0.041	0.910	0.936	0.349
CDGraph	**1**	0.149	**0.186**	**0.002**	**0.022**	**1**	**0.020**	**0.259**	**0.219**	0.031

Table 5: Summarization of evaluation results ($-0.25 < \rho < -0.125$)

Dataset	Facebook					BlogCatalog				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.276	0.110	1.113	0.439	0.288	0.2	0.065	7.036	6.291	0.092
GSM	0.274	0.147	0.677	0.457	0.031	0.26	**0.015**	3.499	3.446	0.075
EDGE	0.307	0.897	2.424	0.926	0.03	0.262	0.391	0.496	0.269	**0.013**
DiGress	0.375	**0.002**	0.725	0.297	0.259	0.188	0.037	0.887	0.946	0.269
CDGraph	**1**	0.149	**0.186**	**0.002**	**0.022**	**1**	0.020	**0.259**	**0.219**	0.031
Dataset	Twitter					Flickr				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.37	0.294	0.450	0.062	0.46	0.323	0.032	2.268	2.348	1.006
GSM	0.326	0.192	0.530	**0.009**	0.012	0.285	0.009	1.065	1.036	**0.014**
EDGE	0.289	0.111	0.068	0.317	**0.002**	0.319	0.429	0.801	0.696	**0.014**
DiGress	0.188	0.348	0.514	0.422	0.15	0.438	0.011	**0.023**	**0.040**	0.371
CDGraph	**1**	**0.045**	**0.001**	0.409	0.016	**1**	**0.008**	0.056	0.068	0.068

Table 6: Summarization of evaluation results ($0.25 < \rho < 0.5$)

C Additional Experimental Results

C.1 More evaluation results

We present the evaluation results under medium/high positive and negative correlations in Tables 4 to 7.

C.2 Sensitivity tests

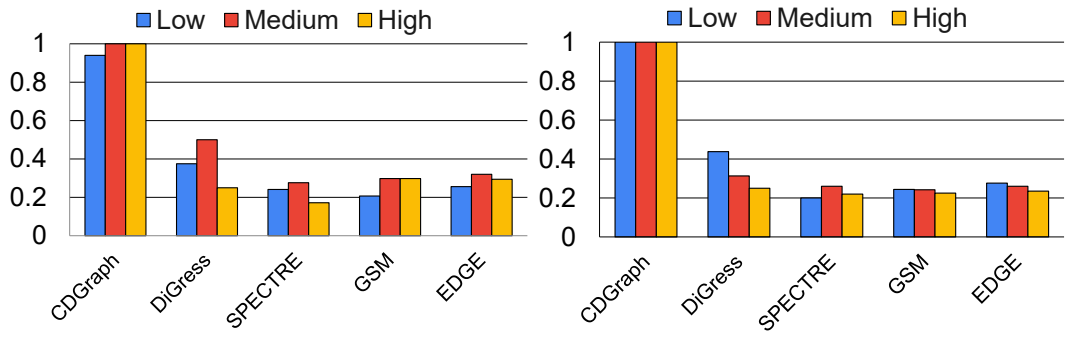
We conduct a sensitivity tests to evaluate the validity of the generated graphs under various degrees of negative correlations ρ of specified conditions. The degrees of negative correlation are low ($-0.125 < \rho < 0$), medium ($-0.25 < \rho < -0.125$) and high ($-0.5 < \rho < -0.25$).

We evaluate the validity under various degrees of negative correlation between specified conditions on Facebook and BlogCatalog in Figs. 5(a) and 5(b), respectively. In both datasets, `CDGraph` is able to achieve higher validity than all baselines under different degrees of negative correlation since it captures the dependencies between nodes with social contagion in the denoising process.

We evaluate the MMD of clustering coefficient under various degrees of negative correlation between specified conditions on Facebook and BlogCatalog in Figs. 6(a) and 6(b), respectively. In both datasets, `CDGraph` is able to generate social graphs with lower MMD of clustering coefficient since the connectivity between nodes can be effectively recovered due to social homophily-based co-evolution in the denoising process.

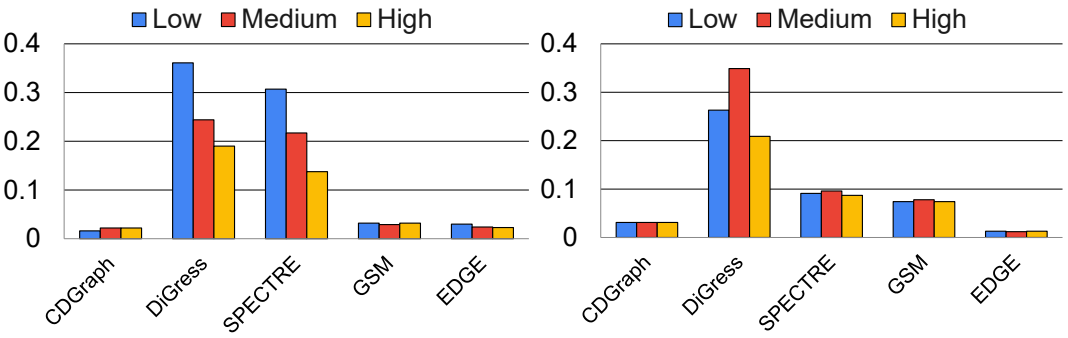
Dataset	Facebook					BlogCatalog				
Metric	Validity	Node	Edge	Density	Clust. coeff.	Validity	Node	Edge	Density	Clust. coeff.
SPECTRE	0.172	0.345	0.312	0.015	0.1376	0.22	0.027	4.879	4.521	0.087
GSM	0.298	**0.076**	0.350	0.337	0.032	0.225	**0.014**	3.487	3.443	0.074
EDGE	0.294	0.817	2.138	0.873	0.023	0.235	0.405	0.544	0.315	**0.013**
DiGress	0.25	0.463	0.653	0.266	**0.19**	0.25	0.133	0.49	0.710	0.209
CDGraph	**1**	0.149	**0.186**	**0.002**	0.022	**1**	0.02	**0.259**	**0.219**	0.031

Table 7: Summarization of evaluation results ($-0.5 < \rho < -0.25$)



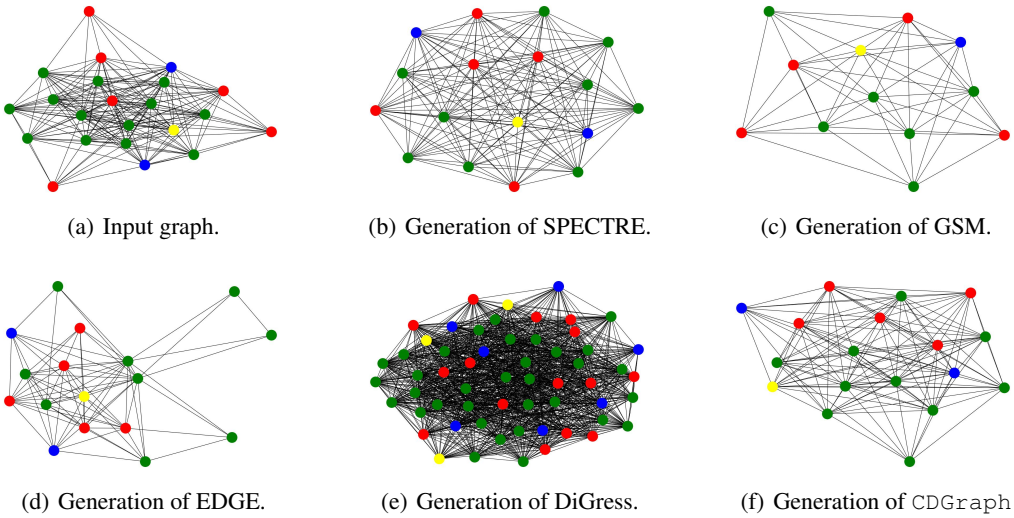
(a) Facebook. (b) BlogCatalog.

Figure 5: Validity under various degrees of negative correlation.



(a) Facebook. (b) BlogCatalog.

Figure 6: MMD of clustering coefficient under various degrees of negative correlation.



(a) Input graph. (b) Generation of SPECTRE. (c) Generation of GSM.

(d) Generation of EDGE. (e) Generation of DiGress. (f) Generation of `CDGraph`

Figure 7: Visualization results of Facebook.

C.3 VISUALIZATION

We present the visualization results of Facebook in Figs. 7. The nodes in green represent that both specified conditions are satisfied.