GRAPH DISCRETE DIFFUSION: A SPECTRAL STUDY

Olga Zaghen[1]*, **Manuel Madeira**[2], **Laura Toni**[3], **Pascal Frossard**[2]
[1] University of Amsterdam, [2] EPFL, [3] Centre for Artificial Intelligence and EEE, University College London
o.zaghen@uva.nl

ABSTRACT

Diffusion models have become central to generative modeling, yet the discrete structure of graphs has prompted the development of specialized approaches and calls for new theoretical tools to understand their dynamics. In this work, we take a step toward addressing this gap by investigating how the noising and denoising processes in the graph discrete diffusion model DiGress manifest in the spectral domain. We first assess whether the learned reverse process faithfully mirrors the forward noise process and analyze the temporal evolution of graph spectra. Our results reveal a gradual shift toward a random graph configuration, with the different noising trajectories depending on the noise model employed. Notably, unlike in image diffusion, we do not observe a clear separation between low- and high-frequency perturbations in the graph spectra. This suggests that alternative perturbation strategies that more effectively leverage the graph spectral domain may warrant further exploration.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have become ubiquitous in generative modeling and are now dominant across many data modalities, with the notable exception of NLP where autoregressive approaches still prevail. Similarly, in graph generation, the prominence of discrete diffusion techniques stems not only from the discrete nature of the data but also from the lack of a canonical node ordering, which makes autoregressive models impractical due to their exponential combinatorial complexity (Niu et al., 2020; Jo et al., 2022).

In continuous domains, such as images, numerous empirical studies have identified progressive small Gaussian perturbations as a crucial component for outstanding performance in diffusion models (Ho et al., 2020; Song et al., 2020). Deeper analyses have sought to explain the success of diffusion models by casting them as spectrally autoregressive models, where generation occurs from low to high frequencies Rissanen et al. (2022). Similar trends have been observed for other continuous data modalities, such as audio, under appropriate transformations Dieleman (2024); Hawthorne et al. (2022); Zhu et al. (2023). In contrast, such a comprehensive understanding is notably absent in graph diffusion. While gaining this insight is crucial for uncovering pitfalls and identifying the key factors that drive performance, it remains a particularly challenging problem. Specifically, because the graph structure changes throughout the diffusion process, the corresponding spectral basis also evolves at each step, making it unclear how to transfer insights from continuous domains and leaving the underlying mechanisms largely unexplored.

In this work, we seek a deeper understanding of graph discrete diffusion by examining how the noising and denoising processes manifest in the spectral domain. To this end, we conduct a detailed analysis of DiGress (Vignac et al., 2022), a well-established model in this setting. Our goal is to characterize its spectral behavior and assess to what extent insights from image diffusion carry over to the graph domain. Specifically, we aim to answer:

1. Does the model-based reverse process accurately reflect the forward noise process, and how do different noise models influence this relationship?
2. What should we theoretically expect regarding graph spectrum perturbations throughout the forward stage (and generative one, if they align)?
3. Do our observations of the forward pass confirm these expectations, and is there any sign of a separation between low- and high-frequency components as observed in images?

*Work done during a research internship in LTS4, EPFL.

Our results indicate that the generative pass generally mirrors the forward process in both spectral and structural features (Section 3). We observe that the graph spectrum gradually converges toward that of a ER graph, with the final density depending on the chosen noise model (Section 4), underscoring the role of the noise model in shaping the generative trajectories. Moreover, while the theoretical predictions for the forward process are confirmed, we do not observe a separation between low- and high-frequency components in the graph spectral domain, contrarily to image diffusion settings (Section 5), highlighting fundamental differences in how graphs are perturbed and opening the door to the potential exploration of spectrum-aware perturbation strategies. Overall, this work presents a first step towards a deeper investigation of the spectral dynamics of graph diffusion, aiming to underscore the importance of such analyses for advancing the understanding and development of graph diffusion models.

2 Background and Problem Setting

In this Section, we describe the setting of our study.

Model. To model graph diffusion in discrete time, we employ DiGress (Vignac et al., 2022). This discrete diffusion model provides a flexible and principled approach to progressively perturb and then denoise graphs, analogous to diffusion models in continuous spaces. In particular, DiGress is based on a D3PM (Austin et al., 2021) adapted for graphs whose nodes and edges have categorical features. In the *forward* process, a clean graph G^0 is gradually corrupted until it becomes fully noisy after T steps (G^T), yielding a trajectory $(G^0, G^1, \ldots, G^T)$. In this work, we consider $T = 500$. In the *reverse* process, a graph transformer network (a type of GNN) is trained to progressively denoise the trajectories generated during the forward process. To overcome the limited representational power of standard GNNs Xu et al. (2018); Morris et al. (2018), graph-theoretic auxiliary features based on cycles and spectral properties are concatenated to the node and edge features, which are then fed to the transformer model at each reverse step.

Noise models. In this study, we focus on edge perturbation to capture structural changes in graphs and therefore do not consider node features. We denote the space of edge attributes by $\mathcal{E}$, of cardinality $|\mathcal{E}|$. For each edge (i, j), $\mathbf{e}_{i,j} \in \mathbb{R}^b$ is its one-hot encoding, and these are collected in a tensor $\mathbf{E} \in \mathbb{R}^{n \times n \times b}$, where n corresponds to the number of nodes. The noise model adopted by DiGress is applied independently to edges via transition matrices. Suppose that the noise on edges is represented by a sequence of transition matrices $(\mathbf{Q}_E^1, \ldots, \mathbf{Q}_E^T)$, where each entry $[\mathbf{Q}_E^t]_{ij} = q(e^t = j \mid e^{t-1} = i)$ denotes the probability of transitioning from state i to state j. Consequently, the categorical distribution that advances the graph one step towards the fully noisy state is given by $q(\mathbf{E}^t \mid \mathbf{E}^{t-1}) = \mathbf{E}^{t-1}\mathbf{Q}_E^t$. The transition matrices take the form $\mathbf{Q}_E^t = \alpha^t \mathbf{I} + (1 - \alpha^t)\mathbf{1}_E \mathbf{v}'_E$, where α^t decreases from 1 to 0 with t following the popular cosine scheduling (Nichol & Dhariwal, 2021), $\mathbf{1}_E \in \{1\}^{|\mathcal{E}|}$ is a vector of ones, and $\mathbf{v}'_E \in \mathbb{R}^{|\mathcal{E}|}$ is a row vector representing the limit noisy distribution of edge types. We consider four noise models: *marginal*, where $\mathbf{v}_E$ is set to the training set's marginal distribution of edge types, promoting transitions towards more prevalent edge types; *uniform*, where $\mathbf{v}_E = (1/|\mathcal{E}|, \ldots, 1/|\mathcal{E}|)$; *absorbing-to-dense*, where $\mathbf{v}_E$ is a one-hot encoded vector with a 1 in the entry corresponding to the existing edge state; and *absorbing-to-sparse*, where $\mathbf{v}_E$ is one-hot encoded with a 1 in the entry corresponding to the non-existing edge state.

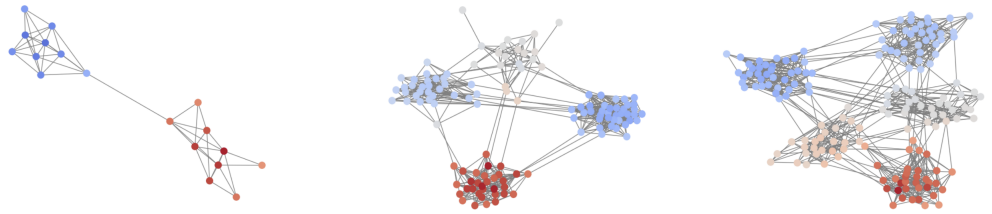


Figure 1: Examples of a Community-small graph (**left**) and of an SBM graph with 4 communities (**center**) and 5 communities (**right**).

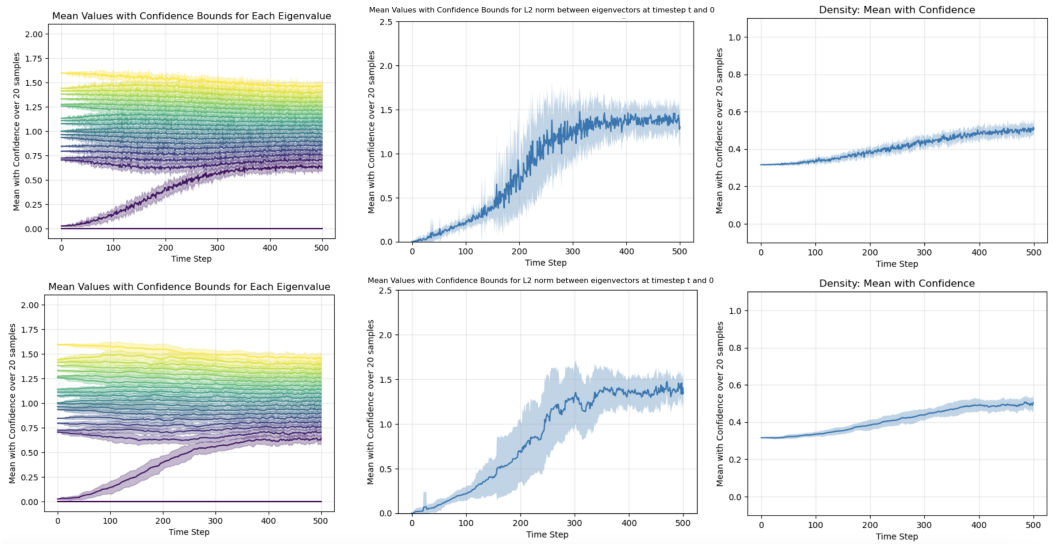


Figure 2: **Top**: Forward pass perturbations of eigenvalues (**left**), the eigenvector corresponding to the first non-zero eigenvalue (**center**), and density (**right**) under a uniform noise schedule for a graph from the Community-small dataset. **Bottom**: Generative pass results for the same quantities. Note that the generative pass proceeds from right ($T = 500$) to left ($T = 0$).

Datasets. The notion of *frequency domain* in the context of graph data refers to the spectral decomposition of the graph Laplacian (or related matrices), where eigenvalues and eigenvectors describe how signals diffuse through the graph. For detailed definition of the Laplacian, see Appendix A.2. Analyzing graphs in this domain can shed light on structural properties, such as the presence of communities. We focus on community-structured datasets, specifically Community-small and Stochastic Block Model (SBM) (Martinkus et al., 2022) (examples in Figure 1), because their clear community organization has a direct spectral interpretation. In particular, the k-th smallest eigenvalue λ_k of the symmetrically normalized Laplacian characterizes how difficult it is to partition the graph into k communities (smaller values indicate a stronger structural division). The spectral gap, i.e., the first non-zero eigenvalue λ_1, specifically highlights the presence of a bottleneck in the graph. In contrast, datasets without clear clusters exhibit less obvious relationships between eigenvalue behavior and changes in connectivity. Further details are in Appendix A.1.

3 Forward vs Generation Pass: a Comparison

In our framework, we use the eigenbasis of the graph Laplacian at each timestep as the frequency domain basis functions. Since node and edge perturbations occur at every timestep, the eigenbasis itself changes accordingly, and even minor structural modifications can lead to significant spectral differences. As a result, it is unclear whether the forward or generative perturbations follow a smooth trend, or even if they align with each other. Both aspects, therefore, merit further analysis.

To compare the forward (noising) and generation passes, we start by analyzing how they affect both spectral and structural features of the graphs. For every graph in the training set and for each timestep $t \in [0, ..., T]$, we generate 10–20 noisy samples and compute relevant metrics. In particular, on the spectral side, we track the eigenvalues of the symmetrically normalized Laplacian and compute the ℓ_2 distance between the eigenvectors at a general timestep $t > 0$ and those at $t = 0$. For the structural analysis, we monitor the evolution of the graph density over time. Together, these metrics enable a detailed comparison between the forward and reverse processes by highlighting when and where discrepancies occur. For a more detailed description of these metrics, see Appendix A. These results are depicted in Figure 2 for community-small graphs and uniform noise distribution. A visual inspection suggests that the forward and generation processes are largely consistent. In Figure 7, in Appendix A.2, we demonstrate that the same trend holds for the other prior distributions.

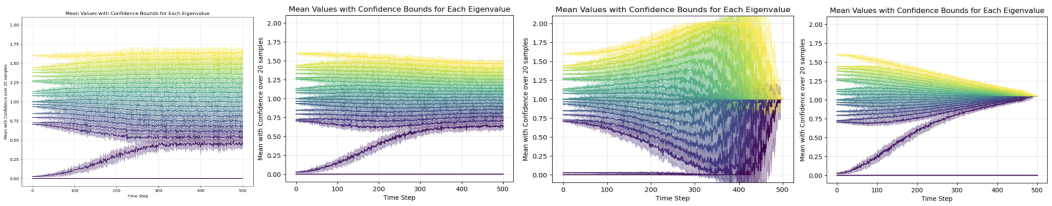


Figure 3: Visualization of eigenvalue perturbation for a 20-node graph from the Community-small dataset under the considered noise models (from left to right): marginal, uniform, absorbing-to-sparse, and absorbing-to-dense, with final densities of 0.3, 0.5, 0, and 1, respectively.

For a more rigorous analysis, we quantify the distributional difference between the forward and reverse processes by computing the Maximum Mean Discrepancy (MMD) between their respective eigenvalue distributions and the distributions of ℓ_2 distances of eigenvectors. The results for the SBM dataset are presented in Appendix A.5. While the discrepancy between both processes is slightly more pronounced when the prior generates very sparse (absorb-to-sparse) or very dense (absorb-to-dense) graphs, our results further reinforce the overall similarity between the forward and generative stages.

4 Theoretical Expectations

The previous section shows that the forward and generation processes exhibit similar patterns. This justifies our focus on the forward diffusion when defining theoretical expectations, since the forward is explicitly formulated, while the reverse is learned by a neural network. Specifically, we are interested in quantifying the convergence rate of an SBM graph toward an *Erdős–Rényi* (ER) random graph by the end of the forward process. In fact, all noise schemes progressively randomize the graph structure, ultimately producing an ER graph with an edge probability set by the noise prior (e.g., $p = 0.5$ for uniform noise). Thus, we expect the spectral configuration of the perturbed graph to match that of an ER graph with the same number of nodes and corresponding edge probability, so that its eigenvalues are uniformly distributed over a symmetric interval around 1. Figure 5 in the center shows an example of ER eigenvalue configuration with 20 nodes.

In an SBM with intra- and inter-community probabilities p and q, the forward process gradually pushes p_t and q_t closer, until they converge to the ER prior-determined value. To assess the rate at which this convergence occurs, we conduct a likelihood-ratio test (LRT) comparing an SBM to an ER model at each timestep. The statistical significance of the likelihood ratio test (LRT) indicates, given the parameters p, q, the number of communities k, and the number of nodes n characterizing an SBM graph, whether the graph can be confidently identified as an ER graph. Testing across different k and n values, we observe that the significance typically rises for $t \in [0.7T, 0.85T]$ (with larger n causing a later increase). Overall, this analysis helps identify the timescale over which the SBM converges to an ER structure. More details on this analysis are provided in Appendix B.

5 Forward Pass Analysis

In this section, given the strong alignment between the forward and reverse processes, we further analyze DiGress's forward process in isolation. Again, given a graph in the training set and for each timestep $t \in [0, ..., T]$, we sample 10-20 forward graph perturbations and compare their distribution of spectral and structural metrics. In particular, for spectral features, we examine the evolution of eigenvalues over timesteps (Figure 3) and extend our study beyond the Community-small and SBM datasets to include Comm-k-NN graphs (see Appendix A.3 for details), allowing us to investigate the behavior of a less conventional type of community-structured graph. We also assess the ℓ_2 distances between eigenvectors over timesteps (Figure 4). Regarding structural features, our primary focus is on graph density.

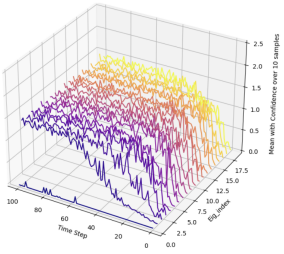


Figure 4: Eigenvector ℓ_2 forward perturbation of a Community-small graph with marginal noise schedule.

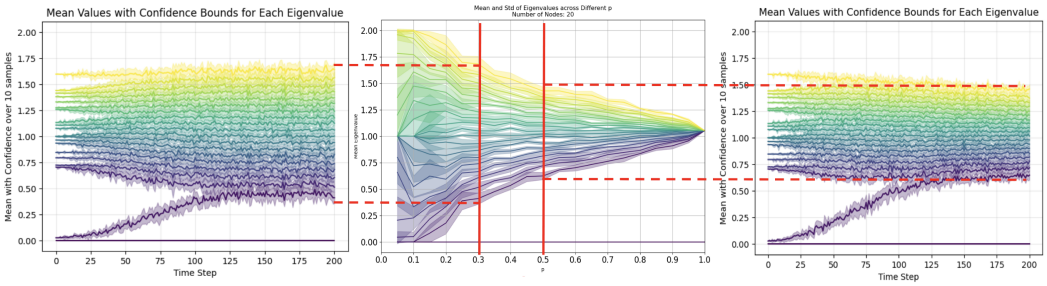


Figure 5: **Left**: Forward-pass eigenvalue perturbation of a 20-node graph from the Community-small dataset under a marginal noise schedule (final $p = 0.3$). **Right**: The same forward-pass perturbation using a uniform noise schedule (final $p = 0.5$). **Center**: Eigenvalue distributions for 20-node Erdős–Rényi graphs at different p-values, with confidence intervals over five samples. In both noise distributions, the dotted lines indicate how the eigenvalues align with those of an ER graph with corresponding p after a certain amount of steps, reflecting the theoretical expectations.

Additionally, we introduce the *Bottleneck Measure* to quantify the graph's *bottleneck-ness* and the *Total Effective Resistance* as a measure of overall connectivity. Both metrics are formally defined and analyzed in Appendices A.2 and A.4. The main takeaways of the observed results are as follows:

1. The density (a value between 0 and 1) converges at the final timestep to the probability defined by the noise model (see Figures 2 and 6). For example, it reaches 0.5 under the uniform noise scheme, while in the marginal case it remains nearly constant, matching the original graphs' edge probability. It increases monotonically in a sigmoid-like pattern, with a concavity change at half the timesteps, possibly due to the use of a cosine noise schedule.
2. For each graph, the eigenvalues and eigenvectors converge to those of an ER graph by $t \approx 0.7T$, confirming our theoretical expectations (see Figures 3 and 5). Although this may seem counterintuitive for the two absorbing distributions shown in Figure 3, only the density, i.e., the ER edge probability, changes after $t \approx 0.7T$ (compare first and second column to the third in Figure 2). This is also evident when comparing the final steps of the absorbing distributions with either the highest or lowest probabilities in the ER plot in Figure 5. Notably, eigenvalues across the entire spectrum undergo perturbations at each timestep, differing from the selective treatment of low- and high-frequency components in image diffusion models (Ho et al., 2020; Rissanen et al., 2022).
3. For each eigenvalue, the perturbation along the noising trajectory depends primarily on its initial deviation from the corresponding ER eigenvalue at the same index. The greater the divergence from the ER configuration associated with the end probability, the more steps are required for stabilization, as reflected in the trends of the eigenvector ℓ_2 differences. For the Community-small and SBM datasets, this initial divergence primarily affects the lowest frequencies, which is a well-known property of community-structured graphs, where the presence of k communities is reflected in the lowest k eigenvalues being close to zero. The less conventional community structure of the Comm-k-NN dataset allows this pattern to also be observed in the highest frequencies.

Finally, note that since the reverse process closely mirrors the forward process, as shown in Section 3, we expect it to exhibit the same spectral trends, naturally in reverse.

6 Related Work

Spectral Analysis on Diffusion Models. Diffusion models (Sohl-Dickstein et al., 2015) have gained widespread adoption for their strong generative performance, prompting deeper analysis of their underlying mechanisms. Early work identified a low-to-high frequency generation process (Ho et al., 2020), which later studies leveraged to enhance data efficiency, latent disentanglement (Rissanen et al., 2022), sampling acceleration, and image manifold characterization (Wang & Vastola, 2023). This coarse-to-fine generation can also be exploited in non-image modalities (Dieleman, 2024). More recently, diffusion models have also been shown to operate across different hierarchi-

cal feature levels throughout their generative process, unveiling the compositional structure of image and text data (Sclocchi et al., 2025; 2024)

Graph Diffusion Models. Similarly to other modalities, diffusion models have become state-of-the-art for graph generation. Early works applied continuous diffusion to graph data (Niu et al., 2020; Jo et al., 2022; 2024), while later approaches introduced discrete diffusion models (Austin et al., 2021; Hoogeboom et al., 2021) to account for the inherent discreteness of adjacency matrices (Vignac et al., 2022; Haefeli et al., 2022). These models rely on Discrete-Time Markov Chains to capture the stochastic diffusion process. While the noise model is crucial to their performance, their connection remains underexplored (Tseng et al., 2023). This work aims to contribute towards bridging this gap.

7 Conclusion

Our findings confirm that the community-based graph structures and their spectral properties gradually converge towards those of a random ER graph, which is consistent with our theoretical expectations. Notably, we do not observe a decoupling of low- and high-frequency eigenvalues over time, contrasting with image diffusion. Instead, the eigenvalues progressively align with those of a random graph, with the most pronounced shifts observed for those that initially differed the most from their ER counterparts. These observations point to the importance of reevaluating both the design of noise models and the choice of perturbation domain. Future work should investigate whether perturbation strategies that appropriately exploit the structure of the spectral domain, rather than operating in the vertex domain, could lead to improved generation quality or control, and extend this analysis to other models and datasets.

8 Acknowledgments

LT acknowledges funding from SRF\R1\231074 Leverhulme Trust Senior Research Fellowships 2023.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *International Conference on Machine Learning (ICML)*, 2023.

Mihai Cucuringu, Xiaowen Dong, and Ning Zhang. Maximum likelihood estimation on stochastic blockmodels for directed graph clustering. In *ArXiv*, 2024.

Karel Devriendt and Renaud Lambiotte. Discrete curvature on graphs from the effective resistance. In *Journal of Physics: Complexity*, 2022.

Sander Dieleman. Diffusion is spectral autoregression, 2024. URL `https://sander.ai/2024/09/02/spectral-autoregression.html`.

Jingqiu Ding, Tommaso d'Orsi, Rajai Nasser, and David Steurer. Robust recovery for stochastic block models. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2022.

Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. In *SIAM review*, 2008.

Kilian Konstantin Haefeli, Karolis Martinkus, Nathanaël Perraudin, and Roger Wattenhofer. Diffusion Models for Graphs Benefit From Discrete State Spaces. In *NeurIPS New Frontiers in Graph Learning Workshop (GLFrontiers)*, 2022.

Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, and Jesse Engel. Multi-instrument music synthesis with spectrogram diffusion. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forr'e, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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 (ICML)*, 2022.

Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. In *International Conference on Machine Learning (ICML)*, 2024.

Xinyu Mao and Jiapeng Zhang. On the power of svd in the stochastic block model. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. Spectre: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In *International Conference on Machine Learning (ICML)*, 2022.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI Conference on Artificial Intelligence*, 2018.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning (ICML)*, 2021.

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

Severi Rissanen, Markus Heinonen, and A. Solin. Generative modelling with inverse heat dissipation. In *International Conference on Learning Representations (ICLR)*, 2022.

Antonio Sclocchi, Alessandro Favero, Noam Itzhak Levi, and Matthieu Wyart. Probing the latent hierarchical structure of data via diffusion models. *International Conference on Learning Representations (ICLR)*, 2024.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. In *Proceedings of the National Academy of Sciences (PNAS)*, 2025.

Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015.

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

Alex M Tseng, Nathaniel Diamant, Tommaso Biancalani, and Gabriele Scalia. Complex preferences for different convergent priors in discrete graph diffusion. In *ICML Workshop: Sampling and Optimization in Discrete Space (SODS)*, 2023.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations (ICLR)*, 2022.

Binxu Wang and John J. Vastola. Diffusion models generate images like painters: an analytical theory of outline first, details later. In *ArXiv*, 2023.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2018.

Ge Zhu, Yutong Wen, Marc-André Carbonneau, and Zhiyao Duan. Edmsound: Spectrogram based diffusion models for efficient and high-quality audio synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A FORWARD AND GENERATION PASS ANALYSIS

A.1 MORE ON PROBLEM SETTING

Datasets. For our analysis we consider community-structured datasets with non informative node features (they are constant), hence we only draw our attention to how the graph connectivity is perturbed, and not the node features. Furthermore, the edge attributes are binary: they just express whether an edge exists or not. The considered datasets have the following properties, and two examples are in Figure 1:

- **Community-small**: 136 graphs, 12-20 nodes each, 2 communities, density 0.3.
- **SBM**: 255 graphs, 2-5 communities, 20-40 nodes per community, density 0.08.

In the analysis of the forward and generation passes, for each graph in the training and generated set and for each timestep $t \in 0, ..., T$, a fixed amount of noisy graphs are sampled (between 10 and 20), and some metrics are computed on them. These metrics are related to spectral or structural features of the graphs, and the average is computed over samples for each graph and timestep, and plotted with the corresponding confidence bound (+/- std). Results are shown in Figures 2 and 7.

For what concerns structural features, we show in Figure 6 additional plots showing density perturbation.

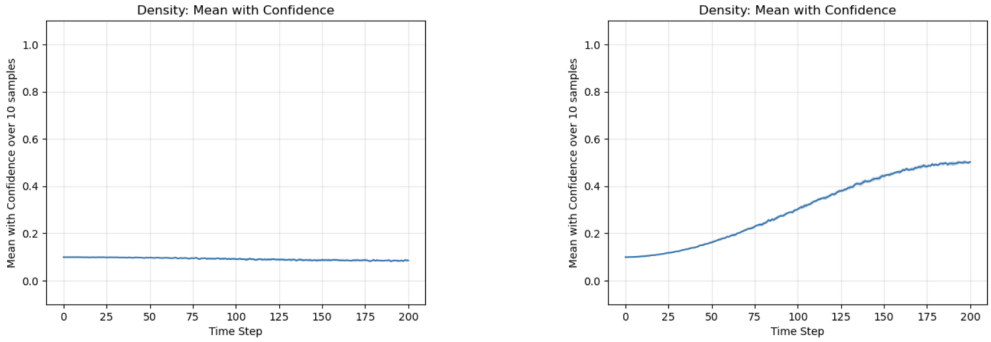


Figure 6: Density perturbation with marginal (**left**) and uniform (**right**) noise schedule across timesteps of the forward pass, computed for a graph in SBM dataset.

A.2 MORE ON SPECTRAL FEATURES

Some useful facts from spectral graph theory that can be useful to keep in mind to better interpret the results are the following:

- All eigenvalues of the symmetrically normalized Laplacian satisfy $0 = \lambda_0, \lambda_1, ..., \lambda_{n-1} \leq 2$, with n being the number of nodes in the graph.
- For the complete graph on n vertices, the eigenvalues are $\lambda_0 = 0, \lambda_i = \frac{n}{n-1}, \forall i > 0$.
- Just as the spectral gap λ_1 measures the obstruction to bipartitioning a graph (the bottleneck), the k-th smallest eigenvalue λ_k is related to partitioning a graph into k parts.

Eigenvalues. For each graph in the input dataset, we computed the symmetrically normalized Laplacian matrix and performed the eigenvalue decomposition of such matrix. The resulting eigenvalues $0 = \lambda_0, \lambda_1, ..., \lambda_{n-1} \leq 2$ take values within the interval $[0, 2] \subset \mathbb{R}$ because of the Laplacian being symmetrically normalized. For each graph, we plotted how all eigenvalues simultaneously vary over timesteps. We observe the following in the forward pass:

- As expected, the lowest eigenvalues, corresponding to the partitions that can be defined on the graph, start from values close to 0 and progressively increase, reaching the eigenvalue

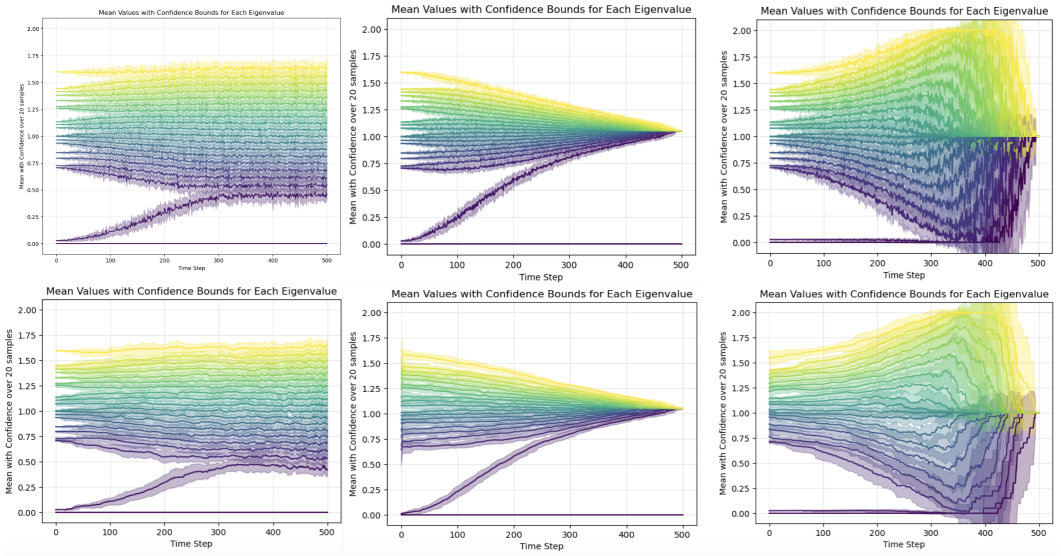


Figure 7: **Top**: Perturbation of eigenvalues of a 20-node graph from Community small with uniform (**left**), absorbing-to-dense (**center**), and absorbing-to-sparse (**right**) noise models. **Bottom**: Corresponding perturbations during the generative pass. All plots depict mean values and standard deviations computed over 20 samples.

configuration that reflects the one of ER graphs with the same amount of nodes and a probability determined by the considered prior distribution.

- A similar trend in the variation of eigenvalues is observed even when setting different values for the total timesteps T.
- All eigenvalues reach the stable configuration of ER graphs at the same timestep, and get perturbed at the same pace in the timesteps previous to that one. We don't observe difference in how high and low eigenvalues get perturbed across timesteps, differently from what is observed between low and high frequencies in diffusion models for images (Scloc-chi et al., 2025).

Eigenvectors. Eigenvectors are not as easy to visualize and interpret as eigenvalues, because of them being n-dimensional vectors and not scalars. In particular, we consider the orthonormal basis of the eigenvectors of the unnormalized graph Laplacian, so that the first eigenvector - the one corresponding to the lowest eigenvalue - is constant, and all other eigenvectors are orthogonal to it.

In order to get an idea of how eigenvectors change over timesteps, we considered, for each eigenvector, the difference between its perturbed version at timestep t and timestep 0, with $0 < t \leq T$. The difference is defined as ℓ_2 norm. We plot, for each graph, how the differences vary over timesteps for all eigenvectors.

We observe the following for the forward pass:

- The ℓ_2 norm plots for all eigenvectors reach a plateau after a certain amount of steps that differs for different eigenvectors.
- As in the case of eigenvalue trends, we observe two different behaviors in how the ℓ_2 norm varies over timesteps, that is determined by whether the eigenvalue/eigenvector pair expresses a potential partition of the graph or not. Specifically, if the graph is composed by k communities, then the eigenvectors corresponding to the eigenvalues with indices $1..k$ are the ones expressing such partitions. These eigenvectors reach stable values (the *plateau*s mentioned in the point before) more slowly than the others, specifically after an amount of steps that are the same needed by eigenvalues to get to the stable configuration of ER graphs. The rest of eigenvectors (except the first one), instead, reach a plateau faster.

Bottleneck Measure on Eigenvectors. In addition to ℓ_2, we have designed another way to measure changes in eigenvectors called the Bottleneck Measure (BM). This new measure aims to better capture how the partition in different communities in the data becomes less clear over timesteps. Since we apply this measure to Community-small, which consists of only two communities, analyzing how the subdivision into communities is affected essentially involves examining how the bottleneck between the two changes. This is why the measure is named as such.

The idea behind this measure is based on the use of an orthonormal basis of eigenvectors. Since the first eigenvector is constant, all other eigenvectors must have entries that sum to zero due to their orthogonality to the first. In fact, if we denote the first eigenvector as $\mathbf{v}_0 = c\mathbf{1}$ and another randomly chosen one as $\mathbf{v}$, then the orthogonality implies $c\mathbf{1} \cdot \mathbf{v} = 0 \rightarrow c\sum_i v_i = 0 \rightarrow \sum_i v_i = 0$. The second eigenvector, which is the first non-constant one, expresses the most natural division of the graph into two communities. In Community-small, where two communities are already present, it has positive values for one of the communities and negative values for the other. We consider the set of node indices in the original graph for which the values were positive (the same would apply if negative values were considered instead). We then compute the sum of these positive values and find the absolute difference with the absolute value of the sum of the corresponding entries at successive timesteps. Given the index set $I := \{j \in \{0, ..., n-1\} \text{ s.t. } x_i^0 > 0\}$ we define it analytically as:

$$\mathbf{BM}(x_i^t) := ||\sum_{i \in I} x_i^0| - |\sum_{i \in I} x_i^t||.$$

We observe that, because the first term in the absolute difference is constant while the second varies with t, the value for BM is 0 for $t = 0$ and reaches its maximum when $|\sum_{i \in I} x_i^t|$ is minimum. In particular, the minimum is reached when the positive entries balance out with the negative ones on the indices I. This happens when the perturbed graph reaches a completely random configuration, in which it's not straightforward to identify two different communities.

We see that the plots for the BM measure are quite similar to those measuring the ℓ_2 norm, with the difference that BM seems to grow more slowly and reach a plateau slightly later in timesteps. An example of this behavior is shown in Figure 8.

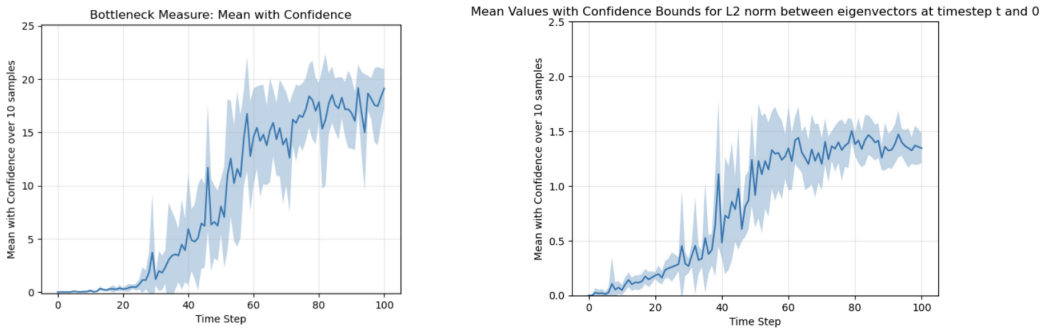


Figure 8: Comparison between the Bottleneck Measure (**left**) and ℓ_2 distance between eigenvectors (**right**) for a 20-node graph from Community-small. We report average and standard deviation over 10 samples, computed during the forward pass with marginal noise schedule.

A.3 Comparison to Different Types of Datasets: Comm-k-NN

The points above are further verified and confirmed by executing the same noise perturbation and plotting the same metrics on different graph datasets. In these, edges are defined with a specific policy (hence not completely random) but differently with respect to SBM. This leads to a different initial eigenvalue configuration compared to the ones observed for Community-small and SBM. These alternative datasets are referred to as Comm-k-NN-intra and Comm-k-NN-inter. We provide their definition in what follows.

Datasets definition. Both datasets consist in an incremental evolution of a base graph G_0, where random edges progressively replace the original graph connectivity. The base graph G_0 consists of

a set of 1500 nodes uniformly divided into $C = 3$ communities. For each node $v_i \in V$ with fixed community membership $c_i \in \{1, 2, 3\}$, we sample its node features $\mathbf{x}_i$ as follows:

$$\mathbf{x}_i \mid c_i \sim \mathcal{N}(\mu_{c_i}, \sigma I)$$

where σ is fixed to 3, and

$$\mu_{c_1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mu_{c_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mu_{c_3} = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

such that the three distributions have large overlapping regions. The edges of the base graph are defined to obtain a k-nn graph, where nodes are connected to their k nearest neighbors inside the same community according to a distance function d, which in our case is the ℓ_2 distance between node features.

Then, starting from G_0, we define a sequence of perturbed graphs $G_1 \dots G_{10}$, where each G_i is obtained from G_{i-1} by randomly removing $i \cdot 10\%$ of edges and randomly adding the same number of edges, thereby maintaining the total amount of edges constant. We employ three distinct policies for this perturbation process:

- **Intra-only edges**: The newly added edges connect only nodes inside the same community.
- **Inter-only edges**: The newly added edges connect only nodes belonging to different communities.

These two policies give rise to the aforementioned Comm-k-NN-intra and Comm-k-NN-inter datasets, respectively.

By examining the perturbation of eigenvalues during the forward pass—as illustrated by two examples in Figure 9—we confirm the observations made in Section 5. In particular, we find that variations in eigenvalue perturbations depend not on the timesteps but on the deviation between an eigenvalue's initial value (for a fixed index) and its corresponding value in the target ER graph. The larger this deviation from the ER configuration determined by the end probability, the more steps are required for stabilization. For the Community-small and SBM datasets, this marked initial divergence is primarily observed in the lowest frequencies, a characteristic of community-structured graphs where the presence of k communities is associated with the lowest k eigenvalues being near zero. In contrast, the less conventional community structure of the Comm-k-NN dataset allows this pattern to also appear in the highest frequencies, as can be observed in this case.

A.4 Total effective resistance

We now provide basic definitions for the concept of effective resistance in a graph and then proceed to explain how we used the concept to conduct measurements and the results we obtained.

The following definitions are partly taken from Devriendt & Lambiotte (2022).

Let's consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, possibly weighted, with associated Laplacian matrix $\mathbf{L}$.

Definition. The *pseudoinverse* of the Laplacian is denoted $\mathbf{L}^\dagger$ and is determined by the equations $\mathbf{L}^\dagger \mathbf{L} = \mathbf{L}\mathbf{L}^\dagger = \mathrm{proj}(\ker(\mathbf{L})^\perp)$ and can be calculated by inverting the nonzero eigenvalues of the Laplacian matrix.

Definition. The *effective resistance* between pairs of nodes is defined based on this pseudoinverse Laplacian as

$$\omega_{ij} := (\mathbf{e}i - \mathbf{e}j)^T \mathbf{L}^\dagger (\mathbf{e}_i - \mathbf{e}_j)$$

for any i, j in the same connected component, where $\mathbf{e}_i$ is the ith unit vector. The resistance matrix Ω is the $n \times n$ matrix containing all pairwise effective resistances, as $(\Omega)_{ij} = \omega_{ij}$.

The concept of effective resistances originates from the theory of electrical circuits, where it captures the resistance exerted by the whole network on a current flowing between any two nodes; more

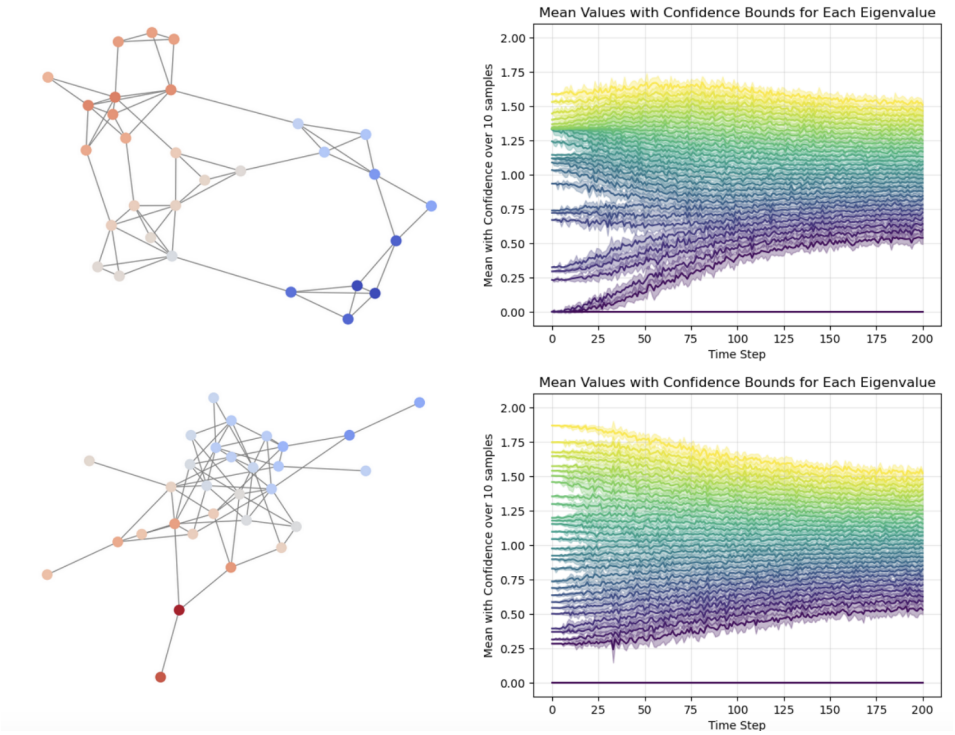


Figure 9: Examples of Comm-k-NN-inter graphs (**left**) and corresponding eigenvalue perturbation (**right**) during the forward pass with marginal noise schedule.

precisely, if a unit current flows through the graph from node i to j, where it passes through the links with resistances $1/c$ and obeys Kirchhoff's laws, then the effective resistance is measured as the voltage difference between i and j (resistance = voltage/current).

The effective resistance has many mathematical properties and one of the most useful ones is that it is a metric between the nodes of a graph:

Theorem. (Devriendt & Lambiotte, 2022) The effective resistance is a metric (a distance) between the nodes of a graph.

Theorem 1 in Devriendt & Lambiotte (2022) provides a good way to think intuitively about effective resistances: if the resistance distance ω_{ij} is large, this reflects that nodes i and j are not well connected in the network – there are few and mainly long (low weight) paths connecting i, j –, whereas a small ω_{ij} means that they are well connected – there are many short paths between i, j. The effective resistance thus reflects a more integrated notion of distance compared to the shortest path distance, since it takes into account all the different paths between two nodes and how they are interconnected.

The *total effective resistance* (the sum of effective resistances over all pairs of nodes in a graph) fully encloses the spectrum of the Laplacian (Black et al., 2023), intuitively representing a scalar measure of how well-connected the network is overall (Ghosh et al., 2008).

The connection between this quantity and the eigenvalues of the network lies in the following equality (Ghosh et al., 2008):

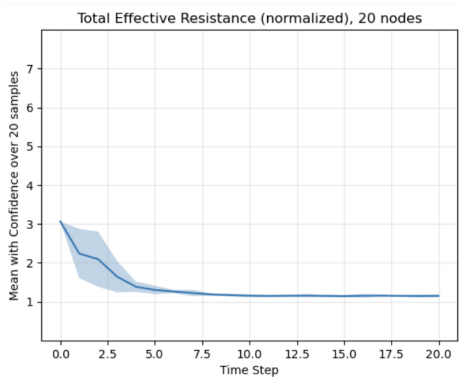
$$R_{tot} = n \sum_{i=2}^{n} \frac{1}{\lambda_i}.$$

Remark. For this equality, the eigenvalues are actually the ones from the unnormalized Laplacian matrix, the matrix that we also considered for analyzing the eigenvector perturbation.

Notably, the total effective resistance of a graph highly depends on the number of its nodes. Therefore, in order to obtain a general measure independent from the amount of nodes in the graph, we considered the normalized total effective resistance of the graphs, that we defined as $R_{tot}^{norm} = \frac{R_{tot}}{(n-1)^2}$. With this kind of normalization, $R_{tot}^{norm} = 1$ for any complete graph independently from the number of nodes. We observe that:

- The higher the prior probability, the closer the final R_{tot}^{norm} is to 1;
- The speed of convergence towards a stable value for R_{tot}^{norm} is high for all noise models.

In Figure 10 we report two examples of the total effective resistance trends.



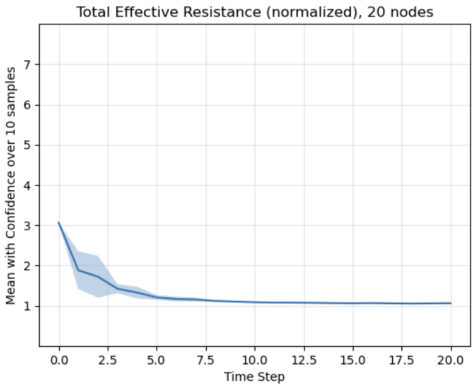


Figure 10: Total effective resistance perturbation across timesteps with marginal (**left**) and uniform (**right**) noise schedule for a 20-node graph from Community-small. We report average and standard deviation over 20 samples.

A.5 MMD Measurements

Relying solely on visual comparison may lack robustness, and it is also particularly difficult to do with large graphs like those in the SBM dataset. To address this, we developed a scalar measure to quantify the difference between the forward and generation pass, for what concerns eigenvalues and ℓ_2 difference of eigenvectors.

To explain the measurement in detail, let us focus on a specific feature we want to measure, such as the eigenvalues or the ℓ_2 difference of eigenvectors. For each timestep and each eigenvalue or eigenvector index, we consider 20 different graph samples in both the forward pass and the generative pass. This gives us 20 eigenvalues or ℓ_2 values for each case. We then assess the difference between the forward and generative passes by calculating the Maximum Mean Discrepancy (MMD) between the 20 values from the forward pass and the 20 values from the generative pass. The MMD is a scalar measure that quantifies the difference between two distributions. Thus, we obtain a scalar result for each timestep and each eigenvalue or eigenvector index.

To create a 2D plot and enhance visualization, we proceed in two ways. First, we can average over the timesteps and plot the results depending on the indices. Alternatively, we can average over all indices and plot the results with the timesteps as the independent variable. Additionally, for a more in-depth analysis, rather than averaging over all timesteps or indices, we first divide their ranges into four sub-intervals and then compute the average. This allows us to separately consider the first, second, third, and fourth sub-intervals, and visualize these four plots together in the same image. This approach enables us to observe differences in behavior between lower and higher eigenvalue or eigenvector indices, as well as variations between low and high timestep values.

Results on Eigenvectors (ℓ_2 norm). When considering eigenvectors, the trends for SBM and Community-small appear to be quite similar, especially when the independent variable is the timesteps (see Figures 11 and 12). In this setting, we observe a higher peak for lower timestep values (closer to clean graphs) that tends to decrease with increasing timesteps, with peaks also in

the higher values in the case of absorbing distributions. In this context, there is no evident sign of a direct correlation between the prior distribution and the difference between forward and generation.

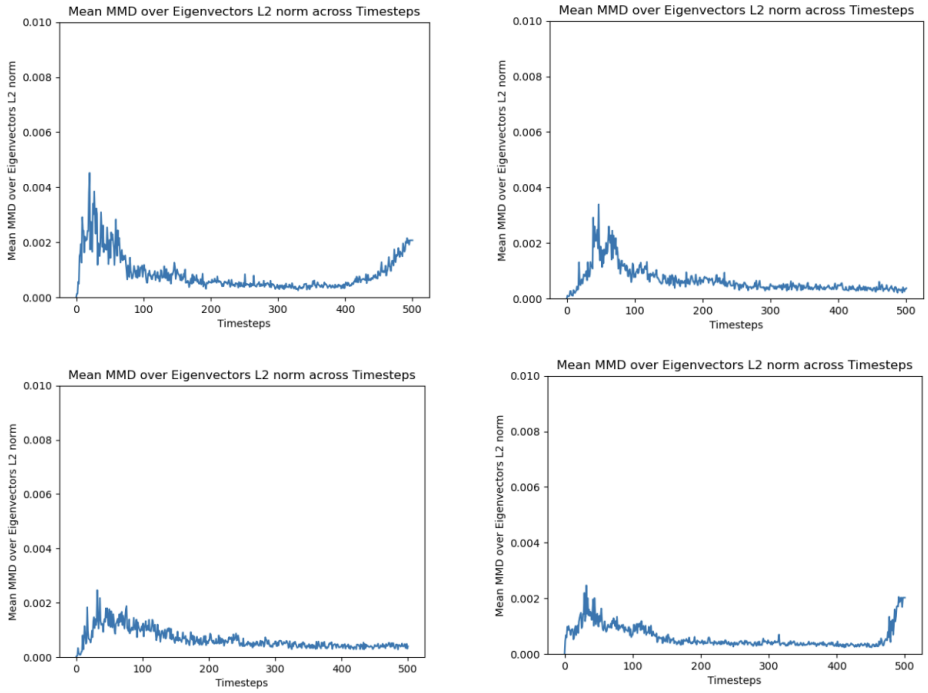


Figure 11: MMD measure of ℓ_2 eigenvector difference with **timestep** as independent variable for a Community-small graph with absorbing-to-sparse (**top-left**), marginal (**top-right**), uniform (**bottom-left**) and absorbing-to-dense (**bottom-right**) noise schedules.

Results on Eigenvalues. When considering eigenvalue trends, there appears to be a stronger dependency on the choice of the dataset. For SBM, there is a direct correlation between the prior distribution and the forward/generation difference. More specifically, we observe the following trends for SBM:

- When the timesteps are the independent variable (Figure 13), the curves decrease as the timesteps increase; moreover, the higher the probability p, the faster the curves decrease.
- When the eigenvalues are the independent variable (Figure 14), we observe higher peaks for low and high eigenvalues. In this case, the higher the prior probability p, the lower the MMD curves trend towards the x-axis.

For Community-small, we can generally observe that the absorbing distributions (dense and sparse) lead to lower MMD values, regardless of whether the independent variable is timesteps or eigenvalue indices (see Figures 15 and 16).

It would be worth investigating further whether the observed dependencies occur by chance due to the specific prior values p chosen for the analysis, or if the trends remain consistent over a wider range of prior probabilities.

B Detailed Theoretical Analysis

Stochastic Block Models are characterized by an intra-community probability p, an inter-community probability q, a number of different communities k, and the total number of nodes n. More generally, inter-community probabilities may vary for different pairs, but we stick to the simplified case corresponding to the SBM dataset we are using with DiGress, in which is it always the same. During the forward diffusion process, the SBM graphs are perturbed, and consequently, p and q are also

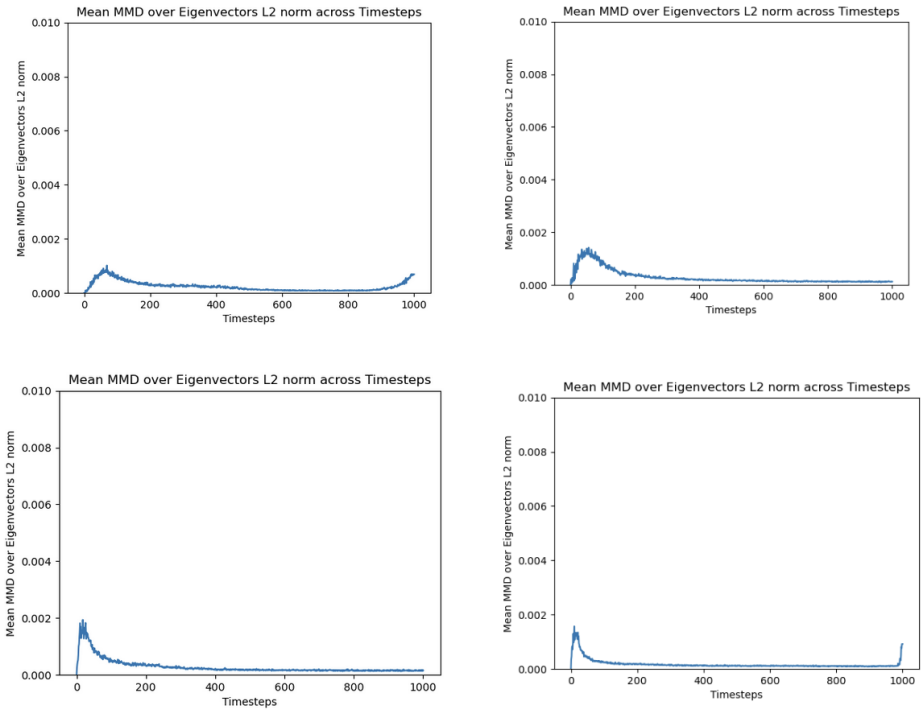


Figure 12: MMD measure of ℓ_2 eigenvector difference with **timestep** as independent variable for an SBM graph with absorbing-to-sparse (**top-left**), marginal (**top-right**), uniform (**bottom-left**) and absorbing-to-dense (**bottom-right**) noise schedules.

progressively altered. We conducted a study with the goal of determining the "similarity" between an SBM model with those varying probabilities and an ER graph.

Regarding the probabilities p and q themselves, we plotted the trend of how they vary over timesteps, from $t = 0$ to $t = T$ when using the cosine noise schedule, with initial values as considered for the SBM dataset in DiGress ($p = 0.3, q = 0.05$) and with marginal transition probability. We see that the variation of the probabilities considered individually depend on the transition matrix, but the difference between the two does not: $p_t - q_t = \bar{\alpha}_t(p_0 - q_0)$. Additionally, they reach the same final value $p_T = q_T$ that is the one determined by the prior.

There exists a specific line of research that aims to analytically characterize the difficulty in discriminating between communities in Stochastic Block Model graphs through clustering techniques. For example, this research derives conditions on SBM parameters to determine how easily they can be distinguished from random graphs (Ding et al., 2022; Mao & Zhang, 2023; Cucuringu et al., 2024).

These works analytically set the conditions for p, q, k and n such that clustering algorithms can identify the different clusters with high probability. This ability of clustering algorithms can be interesting for us because we can think that if the communities are not recognized, then the graph is seen as a random graph. However, this ability to differentiate between the SBM and ER is strictly related to the clustering algorithm itself, which may not be the same for another model that may be used to identify the type of graph.

So, in order to conduct a theoretical analysis independent of the clustering method, instead of using the bounds provided in those papers, we computed a likelihood ratio test between an SBM and an ER model, and then calculated the statistical significance of this statistic.

In short, we did as follows:

- We fixed some values for the total number of nodes and the number of communities (in the example plot in slides, 120 and 4). For each timestep, there are specific values for p and q

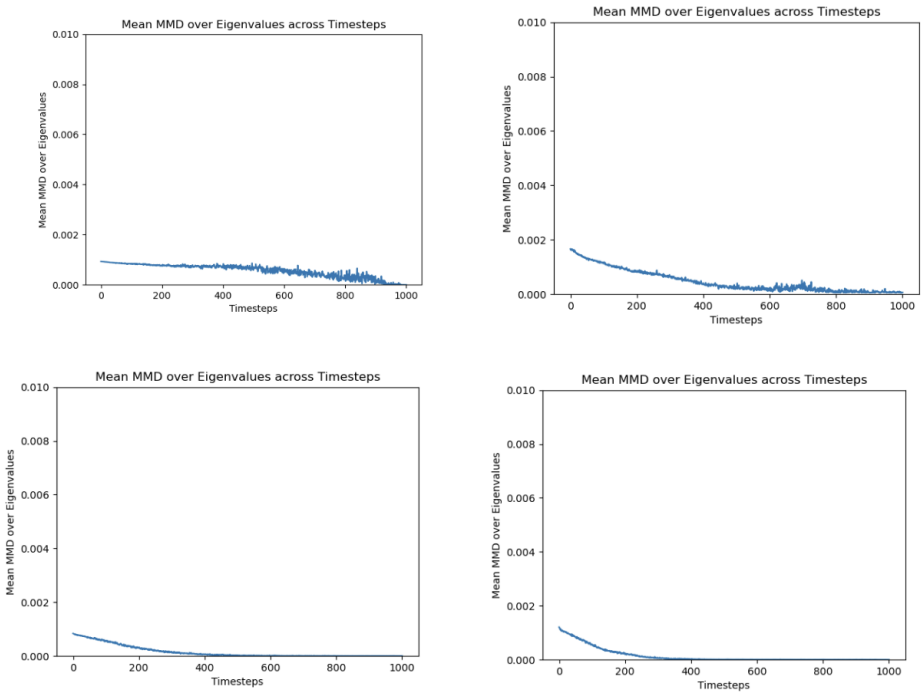


Figure 13: MMD eigenvalue measure with **timestep** as independent variable for an SBM graph with absorbing-to-sparse (**top-left**), marginal (**top-right**), uniform (**bottom-left**) and absorbing-to-dense (**bottom-right**) noise schedules.

shown in the aforementioned plot. With these values, we computed the expected number of inter- and intra-community edges for an SBM graph with these properties.

- Starting from the graph with such expected edges, it's possible to determine the likelihood of this graph being an SBM graph with edge probabilities p and q, and the likelihood of it being an ER graph with edge probability $r = num_edges/all_possible_edges$. More in detail, identifying the adjacency matrix of the considered graph as A, we could compute the two probabilities:

$$P_{\text{SBM}}(A \mid p, q) \;=\; \prod_{1 \le i < j \le n} \left[p^{A_{ij}}\,(1-p)^{1-A_{ij}}\right]^{\mathbb{I}(s_i = s_j)} \;\left[q^{A_{ij}}\,(1-q)^{1-A_{ij}}\right]^{\mathbb{I}(s_i \neq s_j)},$$

$$P_{\text{ER}}(A \mid r) \;=\; \prod_{1 \le i < j \le n} r^{A_{ij}}\,(1-r)^{1-A_{ij}} \;=\; r^{num_edges}\,(1-r)^{all_possible_edges - num_edges}.$$

- We plotted the Log-Likelihood Ratio (LRT) statistic, that is the log of the ratio of the two likelihoods computed as

$$\Lambda \;=\; \ln\left(\frac{P_{\text{SBM}}(A \mid p, q)}{P_{\text{ER}}(A \mid r)}\right) \;=\; \ln P_{\text{SBM}}(A \mid p, q) \;-\; \ln P_{\text{ER}}(A \mid r),$$

  and then we plotted the statistical significance of this statistic (p-value).

The plots obtained through these steps are shown in Figure 17.

The statistical significance should indicate whether, for the considered SBM graph with given p, q, k and n values, it makes sense to identify it as an ER graph, statistically.

By testing with a wider range of node counts (from 50 to 200) and number of communities, the significance always starts increasing at a timestep between 70 and 85 (the more nodes, the later it starts to grow).

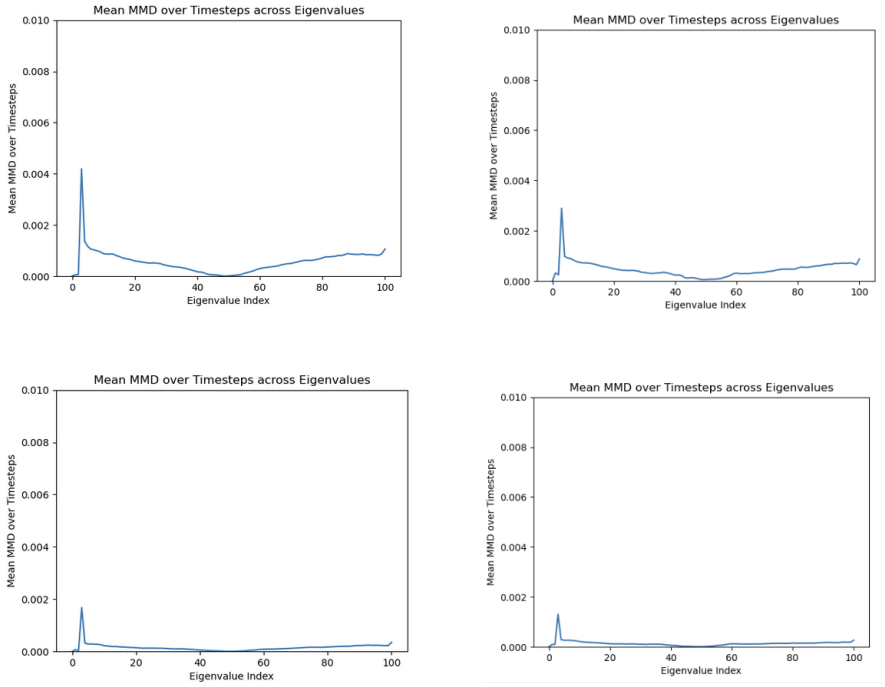


Figure 14: MMD eigenvalue measure with **timestep** as independent variable for an SBM graph with absorbing-to-sparse (**top-left**), marginal (**top-right**), uniform (**bottom-left**) and absorbing-to-dense (**bottom-right**) noise schedules.

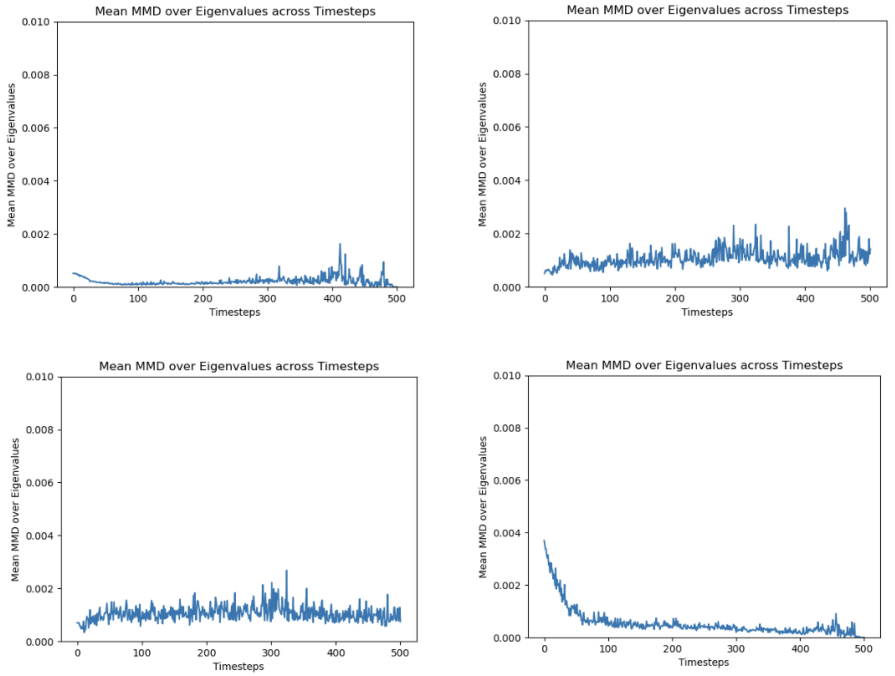


Figure 15: MMD eigenvalue measure with **timestep** as independent variable for a Community-small graph with absorbing-to-sparse (**top-left**), marginal (**top-right**), uniform (**bottom-left**) and absorbing-to-dense (**bottom-right**) noise schedules.

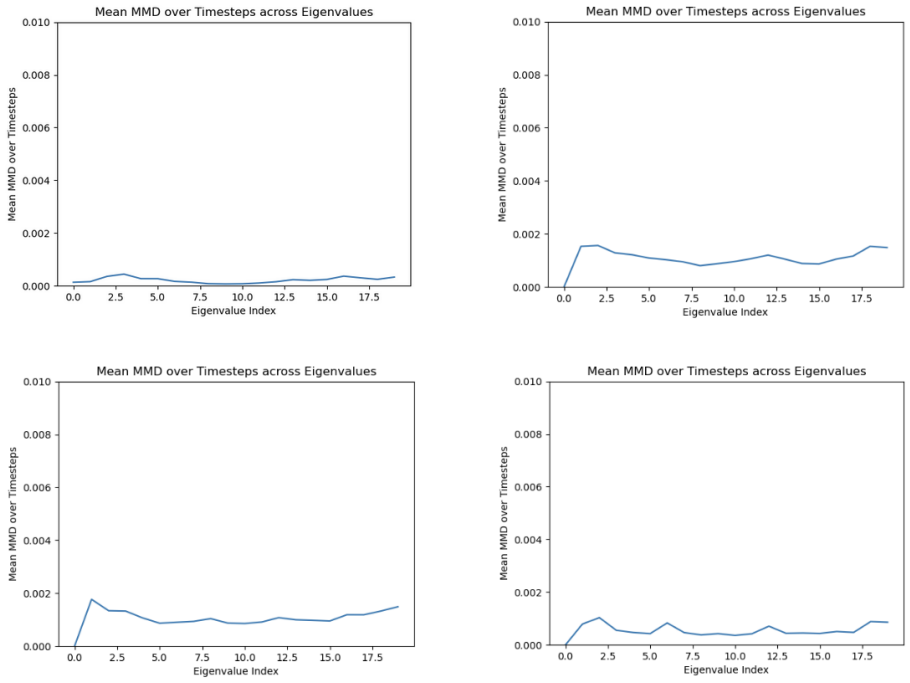


Figure 16: MMD eigenvalue measure with **timestep** as independent variable for a Community-small graph with absorbing-to-sparse (**top-left**), marginal (**top-right**), uniform (**bottom-left**) and absorbing-to-dense (**bottom-right**) noise schedules.

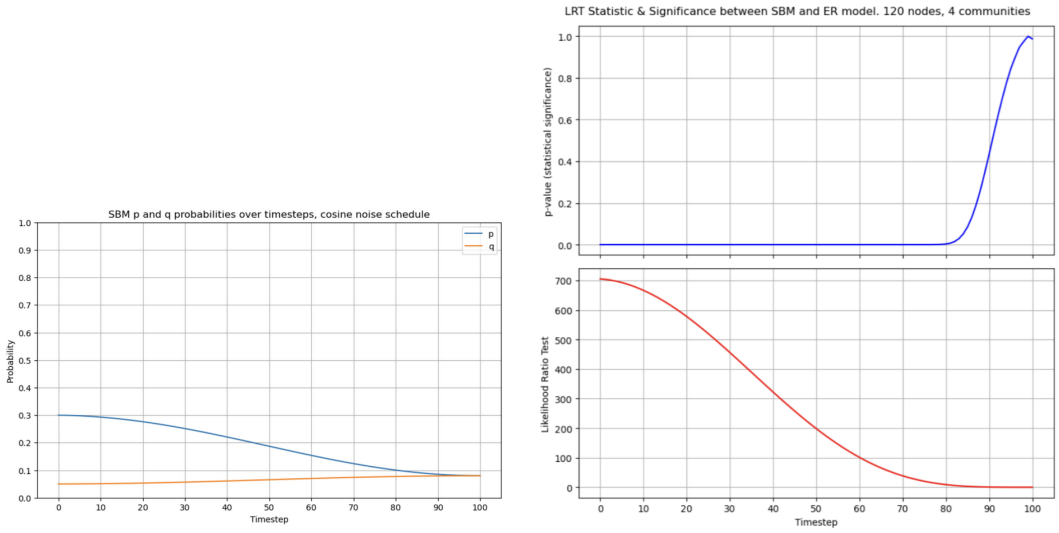


Figure 17: **Left**: Evolution of p and q values over timesteps for an SBM graph perturbed using the marginal model and cosine noise schedule. **Right**: Statistical significance (**top**); Likelihood ratio (**bottom**) for each graph with corresponding p and q being an SBM graph.