TEST-TIME GRAPH EXTRAPOLATION VIA PROGRESSIVE ANCHOR-GUIDED EXPANSION

Abele Mălan[1], **Roberto Gheda**[2], **Robert Birke**[3], **Lydia Chen**[1,2]
[1]UniNE, [2]TU Delft, [3]UniTo `abele.malan@unine.ch`

ABSTRACT

Graph generative models have made significant progress in synthesizing graphs with complex structural properties and strict validity constraints. However, even state-of-the-art models fail to extrapolate to larger graphs. When using models to generate graphs larger than those seen during training, statistical fidelity rapidly deteriorates, and validity approaches zero. Here, we propose Pro-Guid, a test-time extrapolation framework based on structure-aware graph anchoring and conditional synthesis. Starting from a seed graph, the method iteratively selects valid anchor subgraphs and conditionally synthesizes expansion structures that get merged back into the main graph. By constraining the size of anchor graphs and using a repaint-like approach to guide expansion, Pro-Guid avoids overburdening the model with processing overly large structures that degrade generation quality.

1 INTRODUCTION & RELATED WORK

Recent graph generative models enable the creation of synthetic samples with complex structural patterns, including planar, tree, or SBM graphs (Vignac et al., 2023). State-of-the-art models rely on diffusion or related paradigms like flow-matching. While such models excel at generating graphs with node counts observed in their training data, most cannot preserve crucial properties, such as graph-level validity, when generating larger graphs (Bergmeister et al., 2024), despite the inherent ability of neural networks for graphs to operate on arbitrarily large structures.

We tackle node count *extrapolation*, the task of generating graphs larger than those the model was trained on. The main goal of existing graph generators is to maintain statistical fidelity relative to ground-truth graphs from the selected family of the desired size, and to respect the validity criteria characterizing the family. Statistical fidelity is measured based on metrics such as average node degree. Validity is a binary test that denotes a graph's membership in a family (e.g., planar graphs).

Training a diffusion model based on iterative expansion, where the graph's node count grows with each denoising step, helps addressing the shortcomings of classic formulations that fix the number of nodes at the start of denoising (Bergmeister et al., 2024). However, we empirically find that its robustness only covers a limited expansion window, after which generation quality collapses. As the extrapolation ratio grows all models fail to preserve graph validity and even the statistical fidelity is heavily deteriorated. Moreover, extending the benefits of such an approach to other (pre-trained) models would be better served by a sampling-time adaptation.

We propose Pro-Guid to enable highly scalable graph generation in node-number extrapolation settings across different target graph families, and diffusion-based or related models. After synthesizing a seed graph with a node count matching the training distribution, we iteratively grow it to the desired size. At each iteration, we create an anchor based on a subgraph of the already-generated structure, and use guidance to conditionally synthesize a new structure around the anchor with a pre-trained model. We evaluate Pro-Guid across multiple state-of-the-art models, comparing it to the native (unmodified) models, as well as train-time iterative expansion. Code is available online.

2 TEST-TIME GRAPH EXTRAPOLATION

We tackle test-time node count extrapolation for discrete-data pre-trained stochastic interpolants (e.g., diffusion or flow-matching models; detailed in Appendix A) that generate unattributed graphs.

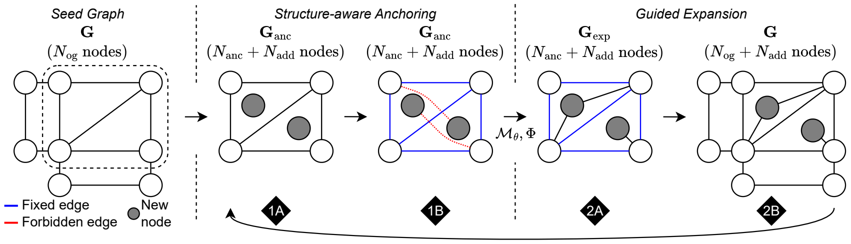


Figure 1: Pro-Guid overview.

Specifically, we aim to use a generator $\mathcal{M}_\theta$ with parameters θ, trained on graphs with at most N_{og} nodes belonging to some family $\mathcal{F}$, to synthesize graphs with $N \gg N_{\text{og}}$ nodes from the same family $\mathcal{F}$ without updating θ. Notably, generated graphs should be statistically similar to real graphs of the same size as defined by statistical distributions $\mathcal{S}^m_{\mathcal{F}}(N)$ for various graph properties m dependent on the node count, while abiding to some validity criteria $\mathcal{V}_{\mathcal{F}} : \mathbf{G} \to \{\bot, \top\}$ related to the graph family $\mathcal{F}$. Note that validity criteria are generally non-differentiable step functions (Kuratowski, 1930; West et al., 2001). Numeric properties like betweenness and graphlet counts are also non-differentiable. In the following, we denote the adjacency matrix of an N-node graph $\mathbf{G}$ as $\mathbf{E} \in [0, 1]^{N \times N}$.

Pro-Guid combines *structure-aware graph anchoring*, which finds a subgraph to which a new substructure can be attached without compromising validity, and *guided expansion*, which steers state-of-the-art backbone graph models to generate a new subgraph around the chosen anchor. By splitting the generation into smaller subgraphs, we avoid overburdening the model with overly large graphs allowing it to maintain validity and statistical properties across the final graph. Figure 1 visualizes the flow of our method, starting with an N_{og}-node graph $\mathbf{G}$ sampled from the model in standard fashion on the left-hand side. The middle part presents the two sub-steps of structure-aware anchoring. They determine the anchor nodes (in white) plus N_{add} new isolated nodes (in gray), around which to expand the current graph and stipulate which edges to preserve (in blue) or forbid (in red) to ensure validity. The right-hand side shows the two sub-steps of guided expansion. We guide the model into connecting new nodes to the anchor and to each other. The resulting new structure is attached to $\mathbf{G}$, completing one expansion iteration. The process is then repeated until reaching the target size. Pro-Guid pseudocode is in Appendix B.

2.1 STRUCTURE-AWARE GRAPH ANCHORING

The first step is to identify a subgraph acting as the anchor from which to expand the structure. Choosing a suitable subgraph is crucial for achieving the expected statistical graph properties and keeping validity. For this, Pro-Guid relies on two selector functions: one for nodes and one for edges. The **node selector** (1A in Figure 1) first picks a subset of nodes from $\mathbf{G}$ defining a subgraph $\mathbf{G}_{\text{sg}} \subset \mathbf{G}$ of size $N_{\text{sg}} \ll N_{\text{og}}$. $\mathbf{G}_{\text{sg}}$'s nodes are picked such that they form a connected structure, ensuring coherence between the current graph and its expansion. Its choice is further informed by the graph family $\mathcal{F}$ at hand, as described in Appendix C. Finally, the function adds N_{add} isolated nodes and returns the result as $\mathbf{G}_{\text{anc}}$. The isolated nodes are the foundation for expansion and dictate how much the existing graph grows during expansion. N_{add} is automatically set to be close to N_{og}, while ensuring that expansion will not overshoot the target node count N. The number of nodes N_{sg} is configurable, but it should be upper-bounded so that $N_{\text{sg}} + N_{\text{add}}$ does not exceed the maximum node capacity N_{og} of the generator. This helps generate valid subgraphs that are statistically similar to the given graph family. Simultaneously, it should be lower-bounded by the minimum anchor size needed to guarantee cohesive expansion. In practice we bound $3 \leq N_{\text{sg}} \leq N_{\text{og}}/2$, meaning that synthesizing a graph with $x * N_{\text{og}}$, requires $\approx x$ to $\approx 2x$ expansions.

The **edge selector** (1B in Figure 1) instead filters the edges, creating a mask $\mathbf{M}$ from the adjacency $\mathbf{E}_{\text{anc}}$ of $\mathbf{G}_{\text{anc}}$ to distinguish between three types of entries: i) *fixed edges* which are the existing edges satisfying global graph validity (val. 1); ii) *forbidden edges* which would violate graph validity at the global structure level (val. -1); and, iii) *candidate edges* that can be determined during graph expansion (val. 0). This mask guides the graph expansion step. After expansion, fixed edges should still be set, while no forbidden edges should exist. Forbidden or fixed edges can be set based on $\mathcal{V}_{\mathcal{F}}$.

Scale	Backbone	Sampling	Planar ($N = 64$)		Tree ($N = 64$)	
			V.U.N. ↑	MMD Ratio ↓	V.U.N. ↑	MMD Ratio ↓
1×	*GraphLE*	Original	0.975	2.7177	1.0	8.3516
	DeFoG	Original	0.975	1.9722	0.95	1.6059
2×	*GraphLE*	Original	**0.975**	**10.2552**	**1.0**	3.1052
	DeFoG	Original	0.05	24.2907	0.0	2.6050
		Grad-Guid	0.15	15.6444	0.0	3.9529
		Pro-Guid	**0.975**	12.2404	0.85	**0.9275**
4×	*GraphLE*	Original	0.775	**28.2908**	**1.0**	1.3033
	DeFoG	Original	0.0	595.8078	0.0	160.9623
		Grad-Guid	0.0	504.6361	0.0	159.783
		Pro-Guid	**0.9**	34.9569	0.85	**0.9275**
8×	*GraphLE*	Original	0.45	44.6768	1.0	2.6247
	DeFoG	Original	0.0	478.2095	0.0	880.9003
		Grad-Guid	0.0	223.0892	0.0	882.6784
		Pro-Guid	**0.825**	342.6815	0.575	**1.5713**

Table 1: Validity and MMD Ratio results for the Planar and Tree datasets.

Backbone	Sampling	Scale					
		1× ($N_{\text{base}} = 141$)		2×		4×	
		V.U.N. ↑	MMD ↓	V.U.N. ↑	MMD ↓	V.U.N. ↑	MMD ↓
GraphLE	Original	0.5	50.6903	0.15	62.7304	0.3	42.8616
DeFoG	Original	0.925	7.3024	0.675	88.8429	0.0	238.2664
	Grad-Guid	N/A	N/A	0.6	88.8429	0.0	238.2664
	Pro-Guid	N/A	N/A	**0.95**	**75.8919**	**0.9**	**137.3592**

Table 2: Validity and MMD Ratio results for the SBM dataset.

Formally, an expansion $\mathbf{G}_{\text{ext}}$ over $\mathbf{G}_{\text{anc}}$ and $\mathbf{M}$ must obey:

$$\mathbb{I}(\mathbf{M} = \mathbf{1}) \odot \mathbf{E}_{\text{ext}} = \mathbf{1}, \quad \mathbb{I}(\mathbf{M} = \textbf{-1}) \odot \mathbf{E}_{\text{ext}} = \mathbf{0} \tag{1}$$

where $\mathbb{I}$ is the indicator function. We require implementations of the node and edge selectors to jointly guarantee that merging $\mathbf{G}_{\text{ext}}$ into $\mathbf{G}$ maintains validity for the current graph family $\mathcal{F}$:

$$\mathit{Equation}\ (1),\ \mathcal{V}_{\mathcal{F}}(\mathbf{G}) = \top,\ \mathcal{V}_{\mathcal{F}}(\mathbf{G}_{\text{ext}}) = \top \quad \Longrightarrow \quad \mathcal{V}_{\mathcal{F}}(\mathbf{G} \cup \mathbf{G}_{\text{ext}})\ \forall\ \mathbf{G}, \mathbf{G}_{\text{ext}}$$

2.2 Guided Graph Expansion

During expansion, we use the generation model to choose which candidate edges to add around the anchor $\mathbf{G}_{\text{anc}}$ (2A in Figure 1) and complete the procedure by merging the new nodes and edges into $\mathbf{G}$ (2B in Figure 1). Since the backbone model is trained to generate graphs from scratch, Pro-Guid needs to steer the model via a guidance mechanism to account for the requirements mandated by $\mathbf{M}$ (which include fixing $\mathbf{G}_{\text{anc}}$'s edges). The guidance procedure should minimize, and ideally nullify, the number of mismatched fixed and forbidden edges in $\mathbf{G}_{\text{exp}}$ relative to $\mathbf{M}$:

$$\|(\mathbb{I}(\mathbf{M} = -1) \odot \mathbf{E}_{\text{ext}}) + (\mathbf{1} - \mathbb{I}(\mathbf{M} = 1) \odot \mathbf{E}_{\text{ext}})\|$$

Moreover, $\mathbf{G}_{\text{exp}}$ should be valid and maintain similar properties to generation from scratch.

To guide generation, we use a repaint-like (Lugmayr et al., 2022) approach. Steering guidance with repainting involves overwriting, possibly multiple times, part of the model's clean prediction $\hat{\mathbf{E}}_0$ with (a corrupted version of) the reference data $\mathbf{E}_{\text{imp}}$ based on $\mathbf{M}$. In many discrete models, the prior p_0 represents random graphs with the same edge density as the target graph family. Consequently, $\mathbf{E}_{\text{ex}}$ is already a plausible candidate for a corrupt input at any time step t by construction, allowing to skip its explicit corruption before overwriting $\hat{\mathbf{E}}_0$. Moreover, as detailed in Appendix D, a single repainting step already produces high-quality results, with little to no degradation in metrics compared to generation from scratch. Consequently, we replace the model's prediction $\hat{\mathbf{E}}_0$ by:

$$\hat{\mathbf{E}}_0' \leftarrow \mathbb{I}(\mathbf{M} = 0) \odot \hat{\mathbf{E}}_0 + \mathbb{I}(\mathbf{M} = 1)$$

3 Evaluation

Backbone Models We test baselines with the state-of-the-art model model DeFoG (Qin et al., 2024) as a backbone. Furthermore, we compare our performance with GraphLE (Bergmeister et al., 2024), a train-time solution robust to generation under node count extrapolation.

Extrapolation Baselines In terms of test-time approaches, we compare Pro-Guid with two baselines: Original and Grad-Guid. *Original* refers to the unmodified backbone. *Grad-Guid* is a baseline that synthesizes the end-to-end and uses gradient-guidance, not for expansion, but to regularize the model's output. Throughout the generation process, we discourage the model from outputting adjacency entries for which the prediction is split evenly between having and not having an edge. Thus, for an adjacency matrix prediction $\hat{\mathbf{E}}_0$ with entries normalized to [0,1], we optimize the loss:

$$\mathcal{L}_{\text{Grad-Guid}} = \|\mathbf{1} - \tanh(|\mathbf{2} \odot \hat{\mathbf{E}}_0 - \mathbf{1}|)\|$$

which peaks for entries close to 0.5 and descends as they approach 0 or 1.

Datasets We consider synthetic network families with complex graph-level validity criteria standard in previous works (Jo et al., 2024; Qin et al., 2024): planar, tree, and SBM. Appendix C includes an overview of the families and their validity criteria. For each dataset, we test multiple extrapolation scales, yielding graphs with $2 - 8\times$ the number of nodes relative to a base value $1\times$ within the training distribution. In the planar and tree datasets, all graphs have $N = 64$, which also gives $1\times$. SBM contains graphs with $N \in [44, 187]$, so we take $N_{\text{base}} = 141$ (the 80%-th percentile) as $1\times$.

Metrics Also, in line with previous work, we measure the V.U.N. and MMD ratios. *V.U.N.* denotes the fraction of valid (respect the validity criteria), unique (not generated multiple times), and novel (not present in the original training data) graphs. Each dataset obeys the validity criterion of the graph family with the same name. *MMD* is the maximum mean discrepancy for a set of graph statistics $\mathcal{S}$ between the synthesized and the test set. We consider the MMD of: average node degree, wavelet transform, eigenvalues of the normalized Laplacian, clustering coefficient, number of 4-node orbits, and betweenness centrality. We report the mean ratio between the MMDs of generated graphs and a reference set sampled from ground-truth graphs of the same size.

Results Quantitative results are in Tables 1 and 2. Appendix E contains qualitative results. DeFoG Original fails to generate valid graphs larger than those in the training set across the board. Its validity drops very close to zero after doubling the graph size in Planar and Tree, and quadrupling it in SBM. Its MMD ratios also grow dramatically. GraphLE maintains good validity and MMD across all Tree scenarios, but suffers significant drops at $4\times$ and, especially, $8\times$ Planar. For SBM, the model struggles even at the base size, but experiences a smaller decline than in the other datasets. Altogether, DeFoG has very poor extrapolation capabilities, continuing the trend previously observed by Bergmeister et al. (2024) on other models. GraphLE fares better, as expected, but its effectiveness often still decays at larger extrapolation factors, despite its extrapolation-aware formulation.

Applying Grad-Guid yields limited improvements over the baseline performance. Given that it relies on gradient optimization and the validity criteria are non-differentiable, Grad-Guid cannot directly optimize validity. That only improves on Planar $2\times$, from 5% to 15%. In most cases, however, MMD improves (especially on Planar) or stays fixed. Gradient-guidance can have positive effects on continuous properties (e.g., node degree), even when optimized indirectly. Contrarily, **Pro-Guid** achieves $> 80\%$ validity in seven out of eight tests. It considerably improves validity, from DeFoG's often 0% rate across all tests, and it matches or surpasses GraphLE's validity in Planar and SBM, while often being competitive on Tree. Validity in Pro-Guid decays exponentially with respect to the scale starting from the $1\times$ value. The slight difference between Planar and Tree at $1\times$ grows considerably at $8\times$. Pro-Guid markedly improves MMD over the Original model and Grad-Guid. However, it often falls behind GraphLE, including Planar $4 - 8\times$, despite leading in validity. Pro-Guid is the best method MMD-wise on Tree, where validity and specific statistical properties, such as node degree and clustering coefficient, are closely related. SBM, which also shows a higher MMD degradation in repainting without extrapolation (Appendix D), has the lowest relative improvement.

4 CONCLUSION

State-of-the-art graph generation models accurately model complex graph families, producing graphs that preserve structural validity constraints and statistical properties. However, when sampling graphs with more nodes than in the training set, validity and statistical fidelity both degrade sharply. The proposed Pro-Guid provides a test-time solution for such extrapolation problems using a pre-trained model, thus avoiding expensive retraining. It uses guidance to expand selected anchor subgraphs starting from a seed graph until reaching the desired size. We show that Pro-Guid maintains high validity and improves statistical fidelity for extrapolation scales of $2 - 8\times$.

REFERENCES

Andreas Bergmeister, Karolis Martinkus, Nathanaël Perraudin, and Roger Wattenhofer. Efficient and scalable graph generation through iterative local expansion. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=2XkTz7gdpc`.

Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=cZTFxktg23`.

Casimir Kuratowski. Sur le probleme des courbes gauches en topologie. *Fundamenta mathematicae*, 15(1):271–283, 1930.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. *CoRR*, abs/2201.09865, 2022. URL `https://arxiv.org/abs/2201.09865`.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. *CoRR*, abs/2410.04263, 2024. doi: 10.48550/ARXIV.2410.04263. URL `https://doi.org/10.48550/arXiv.2410.04263`.

Clément 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=UaAD-Nu86WX`.

Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

A BACKGROUND ON STOCHASTIC INTERPOLANTS

Stochastic Interpolants represent the state of the art for generating graphs. They are defined in terms of two iterative processes: forward and reverse. Internally, these processes represent a graph at iteration $t \in [0, 1]$ using a state space based on adjacency-matrix-like structures $\mathbf{E}_t \in \mathbb{R}^{N \times N}$. The *forward process* transforms, over iterations $t : 1 \mapsto 0$, a clean graph G with adjacency matrix $\mathbf{E}_1$, into a random sample $\mathbf{E}_0$ belonging to a known prior distribution p_0. The *reverse process* spans $t : 0 \mapsto 1$, mapping a random samples $\mathbf{E}_0 \sim p_0$ to a clean adjacency $\mathbf{E}_1$ via $\mathcal{M}_\theta$ and a solver Φ. At each iteration, $\mathcal{M}_\theta$ predicts a clean representation $\hat{\mathbf{E}}_1$ from $\mathbf{E}_t$: $\hat{\mathbf{E}}_1 \leftarrow \mathcal{M}_\theta(\mathbf{E}_t, t)$. Φ then takes a step of size $\Delta t \in [0, 1]$ towards the prediction: $\mathbf{E}_{t+\Delta t} \leftarrow \Phi(\hat{\mathbf{E}}_1, t)$. The final $\mathbf{E}_1$ is the adjacency matrix of the generated graph.

B ALGORITHM OF PRO-GUID

We outline Pro-Guid's pseudocode in Algorithm 1. We start by initializing $\mathbf{G}$ with a seed graph with N_{og} nodes sampled from the generator $(\mathcal{M}_\theta, \Phi)$ at hand (line 2). We then repeatably expand $\mathbf{G}$ until it reaches the target node count N (line 3). Consequently, for $N \leq N_{\text{og}}$ the algorithm stops, making it equivalent to the original sampling. First, we select the anchor $\mathbf{G}_{\text{anc}}$ (line 7) and use it to build a mask $\mathbf{M}$ of restrictions (line 8) for keeping validity during expansion. Then, we use test-time guidance to sample a new graph $\mathbf{G}_{\text{ext}}$ around the fixed anchor (line 10). Finally, to end the iteration, $\mathbf{G}_{\text{ext}}$ is merged into $\mathbf{G}$ (line 11). The runtime of Pro-Guid scales linearly with the extrapolation factor, as the number of anchoring-expansion iterations increases linearly, and the cost of each iteration is upper bounded by the cost of generating a graph with N_{og} nodes.

C GRAPH FAMILIES AND ANCHORING STRATEGIES

Planar graphs are graphs drawable on a 2-dimensional plane without any edges crossing themselves. For such structures, we select an arbitrarily-sized connected subgraph of $\mathbf{G}$ as the anchor.

Algorithm 1 Pro-Guid Sampling

```
Input: target node count $N$, graph family $\mathcal{F}$
$\mathbf{G} \leftarrow$ SampleGraph$(\mathcal{M}_\theta, \Phi, N_{\mathbf{og}})$
while NodeCount$(\mathbf{G}) < N$ do
    {Structure-aware anchoring}
    {Set $N_{\text{pad}}$ to avoid overshooting $N$}
    $N_{\text{add}} \leftarrow \max(N -$ NodeCount$(\mathbf{G}), N_{\text{og}})$
    $\mathbf{G}_{\text{anc}} \leftarrow$ SelectNodes$_{\mathcal{F}}(\mathbf{G}, N_{\text{sg}}, N_{\text{add}})$
    $\mathbf{M} \leftarrow$ SelectEdges$_{\mathcal{F}}(\mathbf{G}_{\text{anc}})$
    {Guided Expansion}
    $\mathbf{G}_{\text{exp}} \leftarrow$ SampleGraphGuided$(\mathcal{M}_\theta, \Phi, \mathbf{M})$
    $\mathbf{G} \leftarrow \mathbf{G} \cup \mathbf{G}_{\text{exp}}$
end while
Return: graph $\mathbf{G}$ with $N$ nodes
```

Anchoring trategy: When masking, we force each new node to be inside a random cycle of the initial subgraph by only allowing it to create edges to the subgraph nodes forming its assigned cycle. In doing so, we maintain planarity under expansion by effectively adding new nodes with connections walled off by the cycles within the existing graph.

Tree graphs are connected graphs with no cycles.

Anchoring strategy: For the anchor, we select an arbitrarily-sized connected subgraph of $\mathbf{G}$, and mask its edges without further consideration. This strategy is suitable since the merger of any two partially overlapping trees is guaranteed to be a tree.

SBM graphs are based on a stochastic block model that assigns each node to a cluster, with one edge density for nodes within the same cluster, and another for nodes between different clusters. We specifically consider the classic case in which intra-cluster density is high and inter-cluster density is low. Moreover, while the validity criterion in our chosen dataset originally includes lower and upper bounds for both the number of nodes per cluster and the number of clusters in the graph, to allow extrapolation to larger node counts, we consider only the lower-bound constraints. Thus, adding more nodes to each cluster beyond the original bounds, adding more clusters within the same range, or a mix of both are all valid strategies. The graphs should, however, still maintain the original inter- and intra-cluster edge probabilities.

Anchoring strategy: We pick the smallest cluster of $\mathbf{G}$ as the anchor and mask only its edges. In doing so, we prioritize creating new clusters with each expansion.

D SINGLE-STEP REPAINTING RESULTS

Table 3 compares unconditional generation to imputation (conditioning on a fixed subgraph) via single-step repainting for graphs with sizes spanning the original distribution. Consequently, for SBM, unlike in the mail evaluation, we generate graphs across the entire distribution of the original dataset, rather than a fixed number of nodes near the top end. The fixed subgraphs are compatible with the family represented by each dataset. Thus, for Planar and Tree, the subgraphs are also planar/tree graphs, while for SBM, the subgraphs represent one or two SBM clusters.

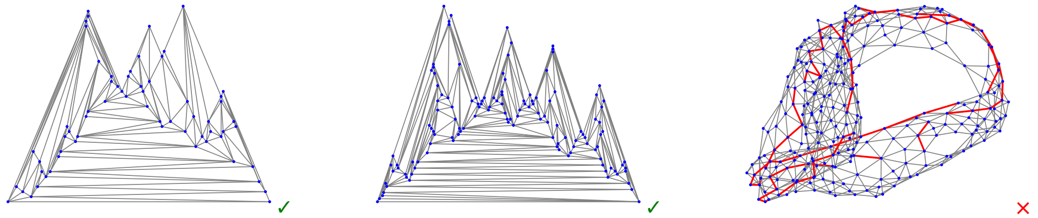
In both the main and imputation experiments, we augment repainting via a self-guidance technique. Namely, in the spirit of classifier-free guidance, we update the entries of the rate matrix, which dictates how to alter the model's clean graph prediction before the next generation step, using a tunable scalar weight W. Classifier-free guidance changes the rate matrix to point towards the output conditioned on the graph-level label and away from the unconditional one. For imputation, we find that, on Planar and Tree, amplifying the direction of the rate matrix for the model's prediction on the repainted data maximizes validity and minimizes deviation from the fixed subgraph.

Dataset	# Imp. Nodes	Imp. Type	Mean MMD Ratio ↓	Validity ↑	Mismatched Entries ↓
Planar (N=64)	0	N/A	1.38	0.975	N/A
	16	Repaint (W=1)	1.48	0.825	2.0
	16	Repaint (W=6)	2.31	0.975	0.2
	32	Repaint (W=1)	3.11	0.875	2.25
	32	Repaint (W=6)	2.68	0.925	0.4
SBM (N=44-187)	0	N/A	2.74	0.925	N/A
	20	Repaint (W=1)	6.37	0.85	4.45
	40	Repaint (W=1)	4.03	0.725	12.4
Tree (N=64)	0	N/A	1.53	0.925	N/A
	16	Repaint (W=1)	6.27	0.85	0.45
	16	Repaint (W=8)	2.04	0.9	0.05
	32	Repaint (W=1)	8.89	0.725	1.45
	32	Repaint (W=8)	7.71	0.95	0.1

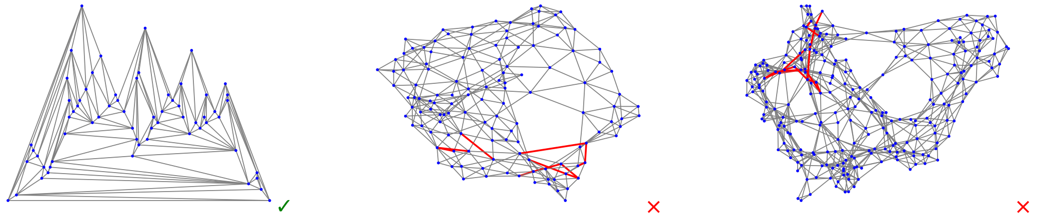
Table 3: Imputation results with DeFoG for 200 sampling steps. *W* denotes the strength of self-guidance applied to the rate matrix in addition to the single-step repainting $W = 1$ is no self-guidance. *Imp. Nodes* gives the size of the fixed subgraph, while *Mismatched Entries* gives the mean number of adjacency entries corresponding to the fixed subgraph that have been altered.

E QUALITATIVE RESULTS

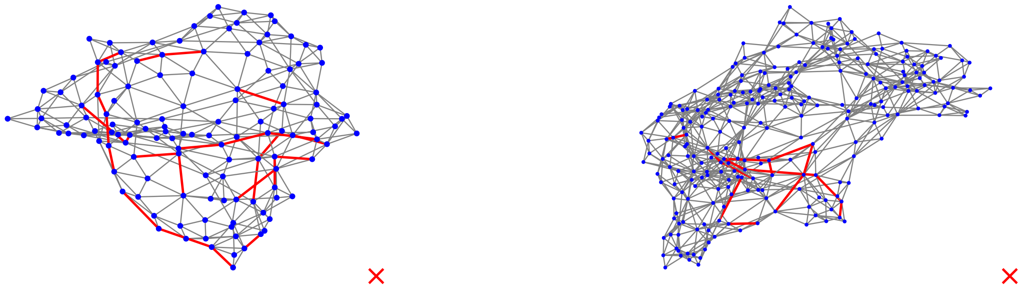
Figure 2 highlights example graphs generated by the different methods at different extrapolation scales on the Planar dataset. We omit $8\times$ extrapolation for visual clarity and highlight the onset of GraphLE's decline in validity at higher node counts via the $4\times$ example.



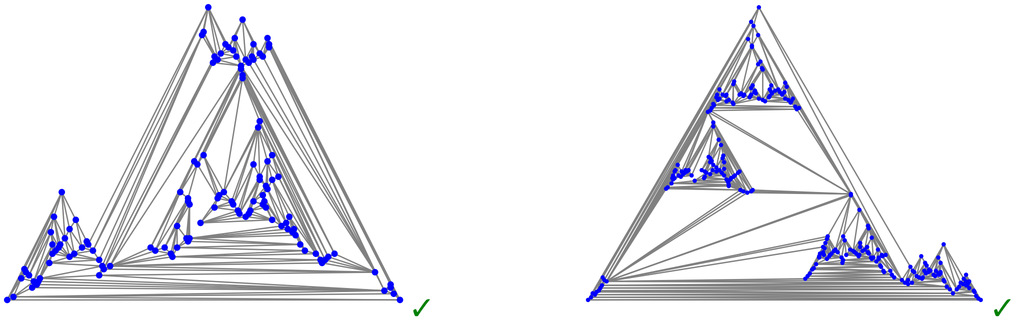
(a) GraphLE $1\times$ ($N = 64$). (b) GraphLE $2\times$ ($N = 128$). (c) GraphLE $4\times$ ($N = 256$).



(d) DeFoG $1\times$ ($N = 64$) (e) DeFoG $2\times$ ($N = 256$) (f) DeFoG $4\times$ ($N = 512$)



(g) Grad-Guid $2\times$ ($N = 282$) (h) Grad-Guid $4\times$ ($N = 564$)



(i) Pro-Guid $2\times$ ($N = 128$) (j) Pro-Guid $4\times$ ($N = 256$)

Figure 2: Qualitative results on the Planar dataset. Valid graphs are marked in the bottom-right corner with a checkmark, while invalid ones are marked with a cross. Red edges indicate a counterexample structure in an invalid (i.e., non-planar) graph.