Topological Graph Generative Model for Ecological Design

Zitong S. Chen
Carnegie Mellon University
samchen@cmu.edu

Oana Carja
Carnegie Mellon University
ocarja@cmu.edu

Abstract

Designing ecological systems, whether for conservation planning, microbial communities, or tissue architectures, requires anticipating how populations will evolve. Yet most existing tools optimize indirect measures that do not reliably predict long-term evolutionary outcomes, limiting their usefulness across domains. We introduce a new approach that treats ecological design as a graph generation problem grounded in evolutionary theory. In this framework, fragmented ecosystems or connectivity networks of interacting individuals in the population are represented as graphs, and we guide their design using two intuitive controls, based on prior theoretical work: amplification, the likelihood that a new trait will spread, and acceleration, the speed at which this happens. To train our model, we create a dataset of 12,173 synthetic networks, each evaluated through evolutionary simulations. We then develop a generative model that proposes designs and a "compile-to-edits" step that translates them into practical, budgeted modifications, while ensuring the network remains connected. Our method exhibits calibrated target to realized control for both factors, uncovers clear structural patterns, produces fixation curves consistent with targets across selection strengths, and outperforms baselines when applied to a real conservation case in the Eldorado National Forest. This work delivers the first end-to-end, evolution-aware generative design tool, advancing principled and budget-conscious ecological interventions.

1 Introduction

As climate, emerging diseases, and human pressures reshape ecosystems, ecological design offers a way to respond. By shaping environments to foster resilience, biodiversity, and sustainability, we can guide how living systems adapt, respond to change and steer them toward desired outcomes. Applications range from constructing habitat corridors that reconnect fragmented landscapes [1–3], to engineering microbial communities that resist invasion [4–6], to modeling tissue architectures that inform cancer treatment strategies [7–9].

Across these biological scales, a consistent theme emerges: population structure and spatial arrangement matter [10–12]. Who interacts with whom, how genes or information flow, and how space constrains these interactions can all tip the balance of ecological and evolutionary outcomes [13–16]. To capture these dynamics, researchers increasingly use networks as mathematical proxies for interaction, reproduction, and replacement [10, 17]. Network-based models have enabled major advances in recent years in understanding how heterogeneous structures shape eco-evolutionary outcomes such as mutant spread, population persistence, and community stability [18–20].

Networks make for a natural and versatile mathematical representation of heterogeneous population structure, because they allow continuous tuning of population complexity. A fully connected network represents a well-mixed population, while regular lattices can model local interactions, and systematic addition or deletion of edges creates a spectrum of intermediate structures. Recent studies have

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Science.

shown that the evolutionary role of such structures can be summarized by two key properties: the amplification factor, which describes how structure alters the probability that a new mutation spreads relative to a well-mixed population [12, 19] and the acceleration factor, which describes how structure alters the time it takes for spread and fixation to occur [12, 18].

Despite these advances, ecological design still lacks practical tools that harness these insights to tailor structure toward specific eco-evolutionary outcomes. Current workflows typically optimize surrogate connectivity metrics (e.g., least-cost paths or circuit theory) [21–24], or coverage targets wrapped in budgeted optimizers [25–28]. While such approaches produce plausible corridors or reserve networks, they target proxies rather than evolutionary dynamics themselves. This often makes solutions brittle across species, landscapes, or movement models, and limits fine-grained control over network topology. The challenge is further amplified by the multi-objective and nonlinear nature of evolutionary control: a structure may increase the chance a beneficial mutation spreads, while simultaneously slowing the rate at which it does so [12, 29]. These trade-offs are difficult to capture with surrogate metrics and costly to simulate directly.

Together, these gaps motivate a generative approach, one that learns to condition on evolutionary targets and produces actionable designs that generalize beyond any single structural property. Advances in graph representation learning and generative modeling have opened new possibilities for design problems where structure is the key lever. Graph neural networks (GNNs) now capture higher-order motifs and dynamics beyond handcrafted indices [30, 31], while diffusion-based generative models have set new benchmarks for controllable generation in molecules, proteins, and other structured domains [32–34]. The latent graph diffusion model (LGD) [35] in particular combines the strengths of autoencoders and diffusion processes, enabling flexible, conditional generation of graphs with desired structural properties. Yet despite this rapid progress, such models have not been applied to ecological design. This gap creates a unique opportunity: to directly learn mappings from evolutionary objectives to actionable network structures that can inform conservation and sustainability interventions.

In this work, we reframe ecological design as a conditional graph generation problem grounded in evolutionary graph theory. Specifically, we make the following contributions:

1. Construct a dataset of 12173 graphs across diverse families and ran Moran birth–death simulations to obtain labels of amplification and acceleration factors.
2. Adapt the latent graph diffusion (LGD) model to our dataset, enabling conditional generation of graphs given target amplification and acceleration.
3. Introduce a post-processing pipeline that translates generated graphs into actionable edit plans under ecological constraints (edge budgets, connectivity preservation).
4. Demonstrate the framework in a corridor-design case study for the Eldorado National Forest, where learned generative control produces strategies that outperform baselines.

Together, these contributions establish the first end-to-end, evolution-aware generative design framework, a principled, data-driven approach that links evolutionary theory to practical, budget-conscious interventions for ecological systems.

2 Methods

2.1 Dataset Curation

We construct a synthetic dataset of 12173 connected graphs to span diverse topologies relevant to evolutionary dynamics. All graphs have $n{=}100$ nodes with target average degree of four. We sample from multiple families and mixtures: Erdős–Rényi (ER), Watts–Strogatz (WS), preferential attachment (PA), k-regular, bipartite, and spatial random geometric graphs (RGG). Family parameters are drawn from broad ranges to diversify clustering, degree heterogeneity, path lengths, and modularity. Graphs are re-sampled until connected (cap 200 attempts) and stored as edgelists. We split train/validation/test by graph instance ($80\%/20\%/20\%$). Distribution of λ and α for this dataset is visualized in Fig. 6 in Appendix.

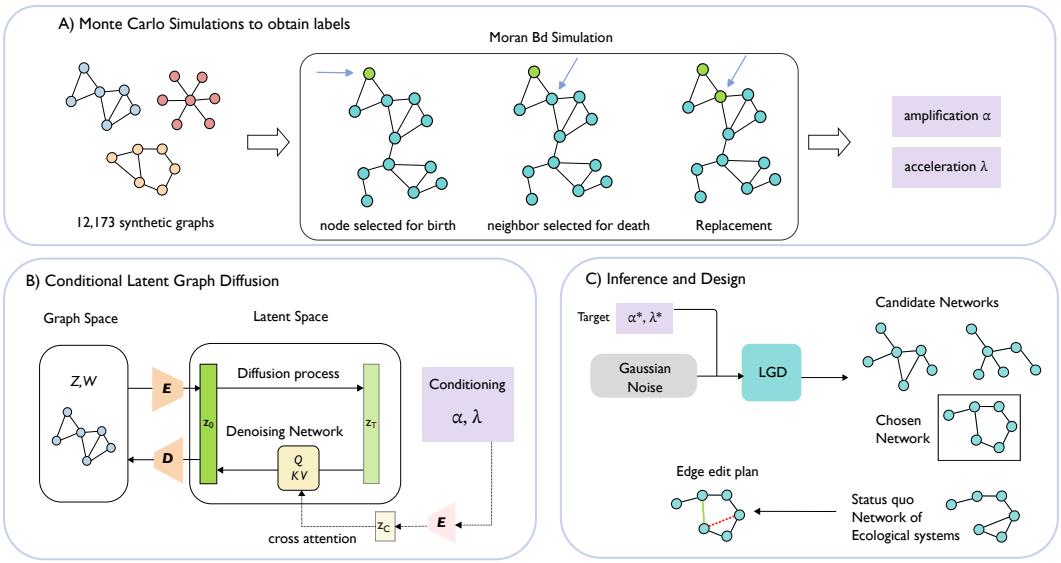

Figure 1: Overview of conditional graph generation for ecological design. (A) Evolutionary simulations on synthetic graphs (with 100 nodes) provide amplification (α) and acceleration (λ) labels. (B) Latent diffusion model projects graphs into latent space and learns to predict noise during the denoising process. Model is conditioned on the α and λ labels obtained through simulations. (C) Conditional sampling produces candidate graphs matching targets. Design layer chooses a single network based on the optimization goal and compiles a feasible edge edit plan under budget constraints.

2.2 Evolutionary Simulation

We simulate the Moran birth–death process on each graph G starting from a single mutant with fitness $1+s$ invading $N-1$ wild types of fitness 1, as is common in population genetic analyses. Let $\Phi_G(s)$ denote the fixation probability on G and $T^G_{\text{fix}}(s)$ the conditional mean time to fixation. The well–mixed reference (complete graph) is denoted $\Phi_{\text{wm}}(s)$ and $T^{\text{wm}}_{\text{fix}}(s)$.

Amplification α. Following the definition in [10, 12], the amplification factor α can be computed by solving

$$\Phi_G(s) \;=\; \frac{1-(1+s)^{-\alpha}}{1-(1+s)^{-\alpha N}}, \qquad \text{(well-mixed is recovered by } \alpha{=}1). \tag{1}$$

Given an estimate of $\Phi_G(s)$, we solve (1) using $s = 0.05$. This value is big enough such that the difference in fixation probability compared to the well-mixed model is sufficiently large, while also small enough to be in the constant regime.

For small s and large N, a first-order expansion yields

$$\Phi_{\text{wm}}(s) = \frac{1}{N} + \frac{N-1}{2N}\, s + \mathcal{O}(s^2), \tag{2}$$

$$\Phi_G(s) = \frac{1}{N} + \frac{N-1}{2N}\, \alpha s + \mathcal{O}(s^2). \tag{3}$$

Hence

$$\alpha \;=\; \frac{\Phi_G(s) - \frac{1}{N}}{\Phi_{\text{wm}}(s) - \frac{1}{N}} \;+\; \mathcal{O}(s), \tag{4}$$

and, when N is large and $s \gg 1/N$ so that the $1/N$ term is negligible,

$$\alpha \;\approx\; \frac{\Phi_G(s)}{\Phi_{\text{wm}}(s)}. \tag{5}$$

For efficiency of model training, we estimate α using Eq. (5) in all of our experiments.

Acceleration λ. Acceleration (deceleration) compares fixation times on G to a well–mixed population:

$$\lambda \;=\; \left[\frac{T_{\text{fix}}^{\text{WM}}(s)}{T_{\text{fix}}^{G}(s)}\right]. \tag{6}$$

For each graph G, we run 20000 Monte-Carlo simulations with initial mutants placed uniformly at random. We record the states at fixation or extinction and calculate α and λ using the probability and time of fixation.

2.3 Conditional Latent Graph Diffusion

We use **Latent Graph Diffusion (LGD)** from [35] as our conditional generative backbone. LGD pretrains a encoder to project graphs into a continuous latent space $H = (Z, W)$, trains a diffusion model in this latent space, and decodes back to discrete graphs. Controllable generation is enabled via cross-attention conditioning on node and edge embeddings. More details about training and inference can be found in Appendix 6.

Conditioning on evolutionary targets. Let $y = (\alpha^*, \lambda^*)$ be the desired amplification/acceleration targets. We follow LGD's conditioning interface: a small MLP embeds y into $\tau(y)$, which is injected into the denoiser via cross-attention over node/edge latents. Training minimizes the standard conditional latent diffusion loss

$$\mathcal{L}_{\text{diff}} \;=\; \mathbb{E}_{H_0,t,\epsilon}\big\|\epsilon_\theta(H_t, t, \tau(y)) - \epsilon\big\|_2^2,$$

with H_t obtained by the forward noising process in latent space. At inference, we sample with DDPM in latent space conditioned on y and decode to graphs.

Encoders used with LGD. Following [35], we pair LGD with a Graph Transformer Encoder [36] to train: (i) an unsupervised structure encoder pretrained by masked reconstruction of node labels and edge existence, used to initialize the latent space, and (ii) a regression encoder trained to predict (α, λ) from a graph (graph-level head only). The structure encoder is also used for label embedding during joint fine-tuning with the diffusion model. The regression encoder is frozen and used during diffusion training for label monitoring, and its loss is solely the graph-level prediction error. The diffusion stage uses only the noise-prediction objective above with no label regression loss added.

2.4 Design Layer: Edge Edits Optimization

Given a baseline network $G_0 = (V, E_0)$ and an LGD proposal $\tilde{G} = (V, \tilde{E})$ conditioned on (α^*, λ^*), we compute an edge-addition plan that moves G_0 toward $\tilde{G}$ subject to a budget constraint. In Fig.5 we use edge additions only; no node edits or deletions are performed.

Feasibility constraint. Let B_e be the edge-edit budget. We also preserve a native connectivity score $C(\cdot)$ at or above baseline by reporting $C(G)$ along the edit path and enforce $C(G) \geq C(G_0)$ at each step.

Objective and scoring. We take as candidates the proposed additions present in $\tilde{G}$ but absent in G_0 ($\mathcal{C} \;=\; \tilde{E} \setminus E_0$). Our optimization objective depends on the goal of our graph design. In the case study of corridor construction in fragmented habitats, the goal is to maximize global efficiency (connectivity) while maintaining λ and α. We therefore rank $(u, v) \in \mathcal{C}$ by the predicted gain in connectivity:

$$s(u, v) \;=\; \widehat{C}\big(G_0 \cup \{(u, v)\}\big) \;-\; \widehat{C}(G_0). \tag{7}$$

Ties are broken by larger degree sum $k_u + k_v$. Scores are recomputed after each accepted edit (greedy re-evaluation).

Greedy edit selection. Initialize $G \leftarrow G_0$ and $\Delta E^+ \leftarrow \emptyset$. Iterate:

1. recompute $s(\cdot)$ on the current G.
2. choose $(u, v)^\star = \arg\max_{(u,v)\in\mathcal{C}\setminus E(G)} s(u, v)$.
3. if $|\Delta E^+| < B_e$ and adding $(u, v)^\star$ keeps $C(\cdot)$ at or above baseline, accept the edit.

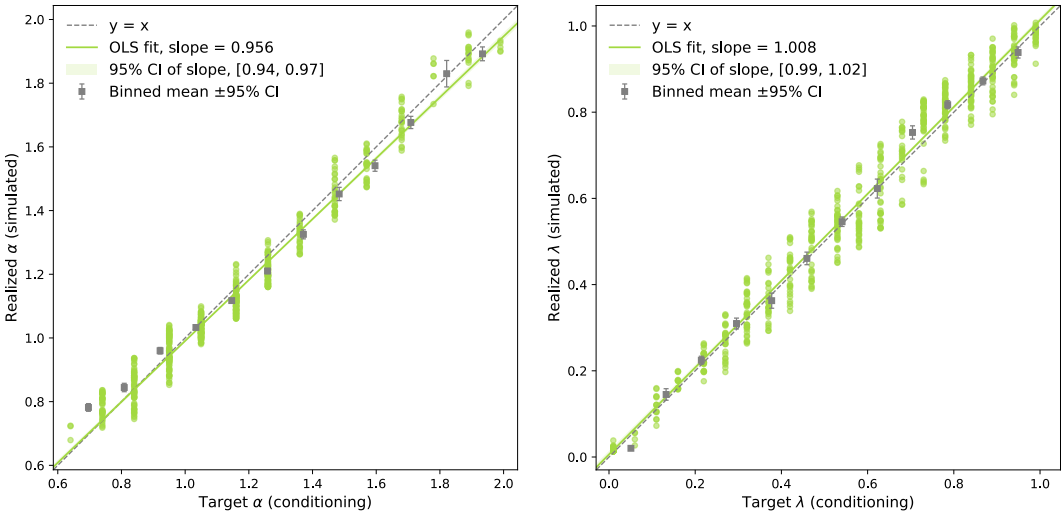

Figure 2: LGD model provides accurate and robust topological control. (Left) Target vs. realized amplification factor (α). We report the slope for the OLS regression, 95% confidence interval, binned mean of the α values, as well as the identity line of $y = x$. (Right) Target vs. realized acceleration factor (λ).

Stop when $|\Delta E^+| = B_e$ or $\mathcal{C}$ is exhausted. The output is an ordered edit list $\Delta E^+ = \{(u, v)_1, \ldots, (u, v)_m\}$ with $m \leq B_e$.

3 Experiments

We evaluate the conditional generator on held-out graphs and a grid of targets (α^*, λ^*). For each target we sample graphs from the model, estimate realized $(\hat{\alpha}, \hat{\lambda})$ via Moran birth–death (Section 2.2), and compute basic structural descriptors (transitivity, assortativity). For plotting, we include graphs for which the realized labels fall within ± 0.1 of the provided targets. Calibration is summarized by OLS slope and mean absolute error (MAE) of target to realized mappings with bootstrap confidence intervals. We also report a success-within-tolerance rate on a coarse (α, λ) grid (tolerances and simulation budgets in the appendix). Our aim is to assess whether a single conditional model learns a generalizable mapping from evolutionary properties to graph topology.

Controllability and calibration. We first evaluate the controllability of the model by looking at the correlation between target labels and realized graph attributes. Results from Fig. 2 summarizes controllability for both evolutionary factors. Target vs. realized amplification (panel A) closely follows the identity line, with an OLS slope of 0.956 (95% CI $[0.94, 0.97]$). Acceleration shows a slope of 1.008 (95% CI $[0.99, 1.02]$) in Fig. 2B which also follows the identity line. We also investigated the range of (α, λ) that the conditional model can successfully generate. Supplementary Fig. 7A shows aggregated success-within-tolerance rate over the (α, λ) grid, indicating a broad band of targets for which the generator repeatedly attains the requested outcomes. Notably, the model generates out-of-distribution for α in $[0.65, 0.8]$ (Fig. 6 left, Fig. 7A). Supplementary Fig. 7B shows that at fixed targets of α^* and λ^*, the model produces diverse solutions of networks while maintaining accuracy of the labels. Together these results show that the conditioning variables act as reliable dials for evolutionary behavior across heterogeneous graph topologies.

Higher-order motifs. We next inspect how graph structure changes as we sweep one target while holding the other fixed (Figure 3). At fixed amplification, increasing λ (acceleration) yields visibly different graph examples (Fig. 3A). As acceleration increases, networks transition from sparse and diffused to increasingly clustered structures. Quantitatively, transitivity rises with λ and exhibits a clear regime change in the mid–λ range (Fig. 3C). Transitivity is the ratio of "closed triplets" (triangles) to all possible triplets in the graph. Higher value of transitivity indicates higher ratio of triangles and more clustered networks. At fixed λ, increasing amplification shifts structure toward hub–leaf organization (Fig. 3B). This trend is reflected in Fig. 3D, where assortativity decreases

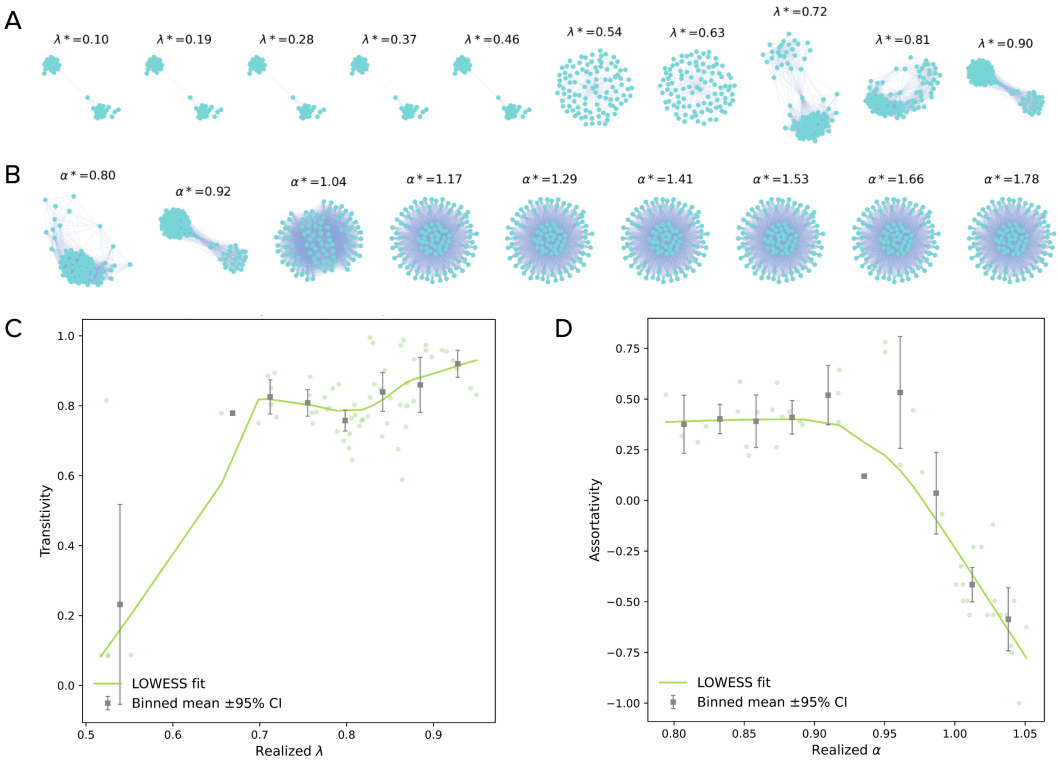

Figure 3: Conditional generation reveals correlation between evolutionary properties and higher order graph motifs. (A) Conditional generation with fixed amplification ($\alpha^* = 1.17$) and increasing λ. (B) Conditional generation with fixed acceleration ($\lambda^* = 0.9$) and increasing amplification (α). (C) Transitivity of generated graphs (fixed $\alpha^* = 1.17$) as a function of realized λ. (D) Assortativity of generated graphs (fixed $\lambda^* = 0.9$) as a function of realized α.

steadily with α. These relationships connect the two evolutionary controls to familiar topological motifs—clustering for acceleration and disassortative hub structure for amplification.

Robustness. Do the generated graphs realize the expected dynamics across selection strengths? Figure 4 answers this by plotting probability and time to fixation across s for representative graphs. With fixed target $\lambda^* = 0.63$, we selected three representative graphs ordered by low/medium/high α^*. These graphs produce cleanly separated fixation-probability curves across a grid of s (Fig. 4A), with higher α yielding higher probability. With fixed target $\alpha^* = 1.29$, graphs ordered by low/medium/high λ produce consistently ordered fixation-time curves (Fig. 4B), with larger λ yielding shorter times. The monotone separations across the entire s sweep indicate that the conditional targets govern dynamics beyond a single s point and supports the ability of the conditional model to be generalized beyond single dataset with fixed selection coefficients.

4 Case study: Corridor Design for Eldorado National Forest

Data. We use the publicly available Eldorado National Forest connectivity dataset hosted on Dryad [37]. The release provides graph-structured connectivity inputs for a parcel network in Eldorado National Forest (California, USA).

Network construction and subsampling. Using custom scripts, we convert the Dryad `.inc` connectivity files into undirected NetworkX graphs. From the parsed network, we work on the largest connected component and extract a connected k=100-node induced subgraph using a two-step "seed + snowball" procedure: (i) choose a seed vertex in the largest component by centrality, and (ii) perform a BFS "snowball" expansion from the seed until k nodes are collected, which guarantees

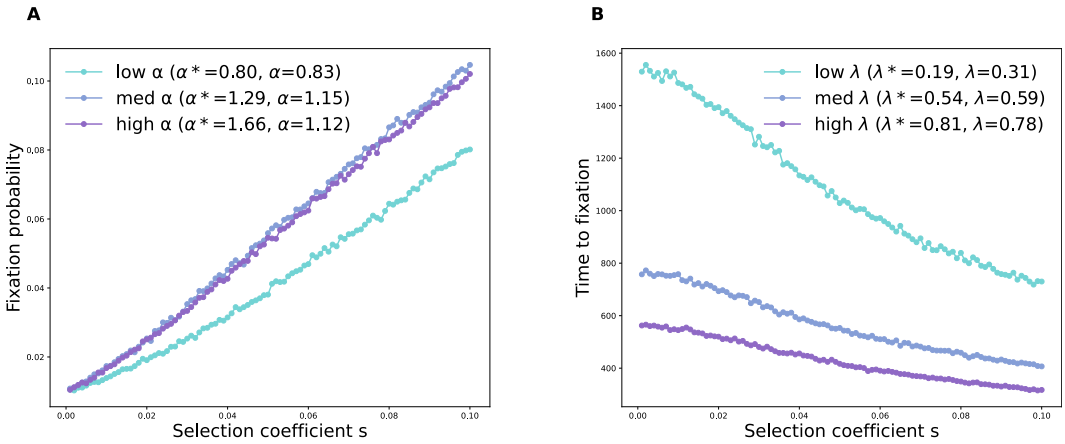

Figure 4: Robustness of simulation results across generated networks. (A) Fixation probability as a function of selection coefficient s for three generated graphs with low, medium, and high amplification (α). Conditioning labels are marked with $*$ while realized labels are not. Target λ is fixed at 0.63. (B) Time to fixation as a function of s for three generated graphs with low, medium, and high λ. Target α is fixed at 1.29.

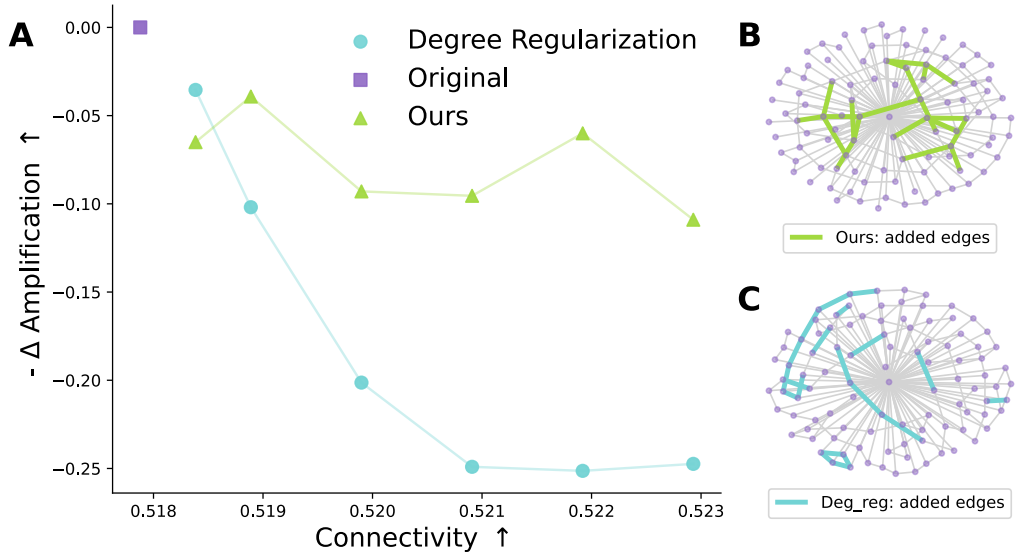

Figure 5: Design under synthetic feasibility constraints. (A) Trade-off between amplification and native connectivity under edge-edit budgets. The goal is to increase connectivity (higher x value) while maintaining the target amplification (higher y value). Comparison shown between our method (green triangles), degree regularization (blue circles), and the original graph (purple square). (B) Example graph edits proposed by our method for edge-edit budgets of 20. (C) Edits proposed from baseline degree regularization method. Added edges are highlighted.

connectivity, then take the induced subgraph on those nodes. This subgraph is the *Original* network G_0 for subsequent analysis.

Baseline evolutionary labels. On G_0 we run Moran birth–death simulations (Section 2.2) to estimate amplification and acceleration: $\widehat{\alpha}(G_0)$ and $\widehat{\lambda}(G_0)$. These values serve as the case-study targets (α^*, λ^*) for conditional generation and for subsequent filtering.

Conditional sampling and filtering. We condition LGD on (α^*, λ^*) and sample candidate graphs. Each candidate $\tilde{G}$ is simulated to obtain realized $(\hat{\alpha}, \hat{\lambda})$. We retain only candidates that (i) match the status-quo labels within tolerance $|\hat{\alpha} - \alpha^*| \leq \delta_\alpha$ and $|\hat{\lambda} - \lambda^*| \leq \delta_\lambda$ ($\delta_\alpha = \delta_\lambda = 0.1$), and (ii) improve the *native connectivity* score $C(\cdot)$ (global efficiency) relative to G_0: $C(\tilde{G}) \geq C(G_0)$.

Compile to edge edits. From a filtered candidate $\tilde{G}$ we compute a feasible *edge-addition plan* that moves G_0 toward $\tilde{G}$ while preserving connectivity (Section 2.4). We evaluate budgets $B_e \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$. The degree-regularization heuristic serves as a baseline for comparison.

Baseline: degree regularization. We compare against a baseline degree-regularization heuristic that prioritizes connecting low-degree nodes. This approach is reasonable as prioritizing least connected nodes is an efficient way of increasing overall connectivity. Specifically, for admissible non-edges (u, v), we rank by

$$r(u, v) \;=\; (\deg_G(u) + \deg_G(v)) + \big|\deg_G(u) - \deg_G(v)\big| \quad \text{(ascending)},$$

and add up to B_e edges greedily while maintaining $C(G) \geq C(G_0)$.

Results. Applying the compile–to–edits procedure on the Eldorado status-quo graph, we evaluate edge-addition budgets $B_e \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$ and report the negative absolute difference of amplification versus native connectivity (Fig. 5A). Across the full budget range, our compiled plans form a higher curve in the (amplification, connectivity) plane than the degree-regularization baseline, and the connectivity of edited graphs remain above the original network. This shows that our model is able to suggest feasible edge edits that improves native connectivity while maintaining the target evolutionary properties. Visualizations in Fig. 5B shows representative edit sets when budget is set to 20. Our plan selects a small set of additions concentrated around bridging connectors, whereas degree-regularization distributes additions along low-degree periphery. Together, these results show that conditional targets can be operationalized as concrete edits that respect the connectivity safeguard while achieving design goals for evolutionary properties under explicit budgets.

5 Conclusion and Future Work

We introduced a general, controllable graph-generation framework that turns evolutionary graph theory into an actionable design tool. Conceptually, the novelty is twofold: (i) we use theory-grounded control variables as the conditioning dial of a modern latent diffusion backbone, and (ii) we complete the loop from targets to concrete edit plans, enabling budget-aware interventions on real networks. This closes a long-standing gap between analytical insights about amplifiers/suppressors and the practical question of "what edges should we add?" Beyond the positive empirical results, the framework matters because it is family-agnostic (not tied to a single graph class), links interpretable structural motifs to evolutionary outcomes, and compiles directly to decisions conservation planners can execute. The same recipe of conditioning on outcome, generating topology, and compiling to edits extends naturally to other structured systems (e.g., invasion management, microbial community, tissue architecture modeling in cancer and AMR). By coupling learned generative control with explicit feasibility constraints, the approach provides a principled pathway for designing ecological networks that are grounded in evolutionary principles and plausible for execution.

Limitations and future work. Our study relies on synthetic training data with a limited range of amplification and acceleration factors. Although the model has demonstrated the ability to generate out-of-distribution to a certain extent, it would still be beneficial to incorporate real-life networks into

the training set to augment the model generalizability. Additionally, the model assumes fixed graph size (n=100) which constrains the application. A natural next step would be to incorporate masking to allow for flexible graph size, and also improve the scalability of the model architecture to allow for generation of larger graphs. Finally, the case study evaluates a connectivity safeguard rather than full landscape costs or policy constraints. Moving forward, our goal is to integrate mapped landscapes with species distribution models, resistance surfaces, and economic/policy costs to allow for more realistic modeling of corridor design.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge support from the NIH National Institute of General Medical Sciences (award no. R35GM147445 to O.C.) and the National Science Foundation CAREER Award (award no. 2442397 to O.C.).

References

[1] Nick M. Haddad, Lars A. Brudvig, Jean Clobert, Kendi F. Davies, Andrew Gonzalez, Robert D. Holt, Thomas E. Lovejoy, Joseph O. Sexton, Mike P. Austin, Cathy D. Collins, William M. Cook, Ellen I. Damschen, Robert M. Ewers, Bryan L. Foster, Clinton N. Jenkins, Andrew J. King, William F. Laurance, Douglas J. Levey, Chris R. Margules, Brett A. Melbourne, A. O. Nicholls, John L. Orrock, Dan-Xia Song, and John R. Townshend. Habitat fragmentation and its lasting impact on Earth's ecosystems. *Science Advances*, 1(2):e1500052, March 2015. doi: 10.1126/sciadv.1500052.

[2] Mayara Guimarães Beltrão, Camila Francisco Gonçalves, Pedro H. S. Brancalion, Ana Paula Carmignotto, Luis Fábio Silveira, Pedro Manoel Galetti, and Mauro Galetti. Priority areas and implementation of ecological corridor through forest restoration to safeguard biodiversity. *Scientific Reports*, 14(1):30837, December 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-81483-y.

[3] Weicheng Sun, Entao Zhang, Yujin Zhao, Zhisheng Wu, Wenhe Chen, Yao Wang, and Yongfei Bai. Conservation priority corridors enhance the effectiveness of protected area networks in China. *Communications Earth & Environment*, 6(1):275, April 2025. ISSN 2662-4435. doi: 10.1038/s43247-025-02227-y.

[4] Stefanie Widder, Rosalind J. Allen, Thomas Pfeiffer, Thomas P. Curtis, Carsten Wiuf, William T. Sloan, Otto X. Cordero, Sam P. Brown, Babak Momeni, Wenying Shou, Helen Kettle, Harry J. Flint, Andreas F. Haas, Béatrice Laroche, Jan-Ulrich Kreft, Paul B. Rainey, Shiri Freilich, Stefan Schuster, Kim Milferstedt, Jan R. van der Meer, Tobias Groβkopf, Jef Huisman, Andrew Free, Cristian Picioreanu, Christopher Quince, Isaac Klapper, Simon Labarthe, Barth F. Smets, Harris Wang, and Orkun S. Soyer. Challenges in microbial ecology: Building predictive understanding of community function and dynamics. *The ISME Journal*, 10(11):2557–2568, November 2016. ISSN 1751-7370. doi: 10.1038/ismej.2016.45.

[5] Chang-Yu Chang, Jean C. C. Vila, Madeline Bender, Richard Li, Madeleine C. Mankowski, Molly Bassette, Julia Borden, Stefan Golfier, Paul Gerald L. Sanchez, Rachel Waymack, Xinwen Zhu, Juan Diaz-Colunga, Sylvie Estrela, Maria Rebolleda-Gomez, and Alvaro Sanchez. Engineering complex communities by directed evolution. *Nature Ecology & Evolution*, 5(7):1011–1023, July 2021. ISSN 2397-334X. doi: 10.1038/s41559-021-01457-5.

[6] Jiliang Hu, Matthieu Barbier, Guy Bunin, and Jeff Gore. Collective dynamical regimes predict invasion success and impacts in microbial communities. *Nature Ecology & Evolution*, 9(3):406–416, March 2025. ISSN 2397-334X. doi: 10.1038/s41559-024-02618-y.

[7] Philipp M. Altrock, Lin L. Liu, and Franziska Michor. The mathematics of cancer: Integrating quantitative models. *Nature Reviews Cancer*, 15(12):730–745, December 2015. ISSN 1474-1768. doi: 10.1038/nrc4029.

[8] Jeffrey West, Ryan O. Schenck, Chandler Gatenbee, Mark Robertson-Tessi, and Alexander R. A. Anderson. Normal tissue architecture determines the evolutionary course of cancer. *Nature Communications*, 12(1):2060, April 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22123-1.

[9] Jennifer C. Ashworth and Thomas R. Cox. The importance of 3D fibre architecture in cancer and implications for biomaterial model design. *Nature Reviews Cancer*, 24(7):461–479, July 2024. ISSN 1474-1768. doi: 10.1038/s41568-024-00704-8.

[10] Erez Lieberman, Christoph Hauert, and Martin A. Nowak. Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316, January 2005. ISSN 1476-4687. doi: 10.1038/nature03204.

[11] Marcus Frean, Paul B. Rainey, and Arne Traulsen. The effect of population structure on the rate of evolution. *Proceedings. Biological Sciences*, 280(1762):20130211, July 2013. ISSN 1471-2954. doi: 10.1098/rspb.2013.0211.

[12] Yang Ping Kuo, César Nombela-Arrieta, and Oana Carja. A theory of evolutionary dynamics on any complex population structure reveals stem cell niche architecture as a spatial suppressor of selection. *Nature Communications*, 15(1):4666, May 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-48617-2.

[13] T. Antal, S. Redner, and V. Sood. Evolutionary dynamics on degree-heterogeneous graphs. *Physical review letters*, 96(18):188104, May 2006. ISSN 0031-9007. doi: 10.1103/PhysRevLett.96.188104.

[14] Andreas Pavlogiannis, Josef Tkadlec, Krishnendu Chatterjee, and Martin A. Nowak. Amplification on Undirected Population Structures: Comets Beat Stars. *Scientific Reports*, 7(1):82, March 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-00107-w.

[15] Suppressors of fixation can increase average fitness beyond amplifiers of selection | PNAS. https://www.pnas.org/doi/10.1073/pnas.2205424119, 2022.

[16] Jakub Svoboda, Soham Joshi, Josef Tkadlec, and Krishnendu Chatterjee. Amplifiers of selection for the Moran process with both Birth-death and death-Birth updating. *PLoS computational biology*, 20(3): e1012008, March 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1012008.

[17] T. Monk, P. Green, and M. Paulin. Martingales and fixation probabilities of evolutionary graphs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 470(2165):20130730, May 2014. doi: 10.1098/rspa.2013.0730.

[18] Yang Ping Kuo and Oana Carja. Evolutionary graph theory beyond pairwise interactions: Higher-order network motifs shape times to fixation in structured populations. *PLOS Computational Biology*, 20(3): e1011905, March 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1011905.

[19] Yang Ping Kuo and Oana Carja. Evolutionary graph theory beyond single mutation dynamics: On how network-structured populations cross fitness landscapes. *Genetics*, 227(2):iyae055, June 2024. ISSN 1943-2631. doi: 10.1093/genetics/iyae055.

[20] Anush Devadhasan, Oren Kolodny, and Oana Carja. Competition for resources can reshape the evolutionary properties of spatial structure. *PLOS Computational Biology*, 20(11):e1012542, November 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1012542.

[21] Claire D. Stevenson-Holt, Kevin Watts, Chloe C. Bellamy, Owen T. Nevin, and Andrew D. Ramsey. Defining Landscape Resistance Values in Least-Cost Connectivity Models for the Invasive Grey Squirrel: A Comparison of Approaches Using Expert-Opinion and Habitat Suitability Modelling. *PLOS ONE*, 9(11): e112119, November 2014. ISSN 1932-6203. doi: 10.1371/journal.pone.0112119.

[22] Brad H. McRae, Brett G. Dickson, Timothy H. Keitt, and Viral B. Shah. Using Circuit Theory to Model Connectivity in Ecology, Evolution, and Conservation. *Ecology*, 89(10):2712–2724, 2008. ISSN 1939-9170. doi: 10.1890/07-1861.1.

[23] Bowen Jin, Jianwei Geng, Zhengning Ding, Linye Guo, Quanquan Rui, Jiamei Wu, Shengqi Peng, Ruichong Jin, Xinwen Fu, Hui Pan, and Guochang Ding. Construction and optimization of ecological corridors in coastal cities based on the perspective of "structure-function". *Scientific Reports*, 14(1):27945, November 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-79433-9.

[24] Yi An, Shiliang Liu, Yongxiu Sun, Fangning Shi, and Robert Beazley. Construction and optimization of an ecological network based on morphological spatial pattern analysis and circuit theory. *Landscape Ecology*, 36(7):2059–2076, July 2021. ISSN 1572-9761. doi: 10.1007/s10980-020-01027-3.

[25] C. R. Margules and R. L. Pressey. Systematic conservation planning. *Nature*, 405(6783):243–253, May 2000. ISSN 1476-4687. doi: 10.1038/35012251.

[26] PMAP - Toolbox detail page for Zonation (Spatial Conservation Planning). https://www.pik-potsdam.de/~wrobel/mediation-platform/toolbox/zonation.html.

[27] Eline S. van Mantgem, Johanna Hillebrand, Lukas Rose, and Gunnar W. Klau. Coco: Conservation design for optimal ecological connectivity. *Frontiers in Ecology and Evolution*, 11, November 2023. ISSN 2296-701X. doi: 10.3389/fevo.2023.1149571.

[28] François Hamonic, Yann Vaxès, Basile Couëtoux, and Cécile H. Albert. GECOT: Graph-based ecological connectivity optimization tool. *Methods in Ecology and Evolution*, n/a(n/a). ISSN 2041-210X. doi: 10.1111/2041-210X.70055.

[29] Benjamin Allen, Gabor Lippner, Yu-Ting Chen, Babak Fotouhi, Naghmeh Momeni, Shing-Tung Yau, and Martin A. Nowak. Evolutionary dynamics on any population structure. *Nature*, 544(7649):227–230, April 2017. ISSN 1476-4687. doi: 10.1038/nature21723.

[30] Thomas N Kipf and Max Welling. SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS. *International Conference on Learning Representations*, 2017.

[31] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, May 2021.

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020.

[33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS. *International Conference on Learning Representations*, 2021.

[34] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant Diffusion for Molecule Generation in 3D, June 2022.

[35] Cai Zhou, Xiyuan Wang, and Muhan Zhang. Unifying Generation and Prediction on Graphs with Latent Graph Diffusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

[36] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, December 2022.

[37] Yicheng Wang, Peng Qin, and Hayri Önal. An optimisation approach for designing wildlife corridors with ecological and spatial considerations. *Methods in Ecology and Evolution*, 13(5):1042–1051, 2022. ISSN 2041-210X. doi: 10.1111/2041-210X.13817.

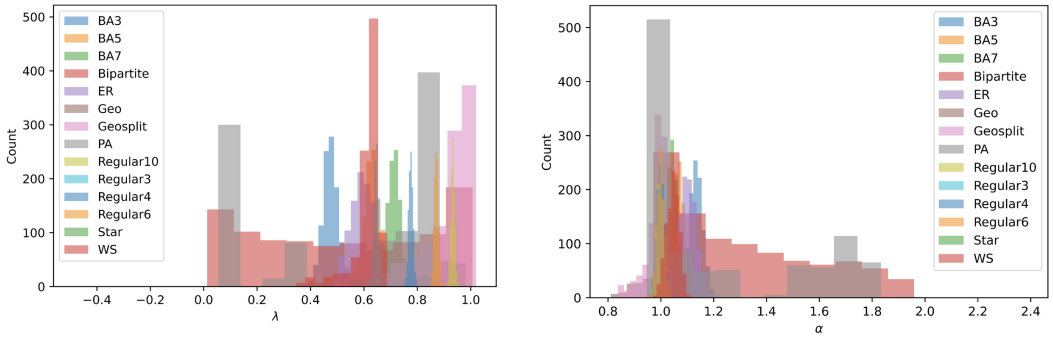

Figure 6: Distribution of λ and α in training data labeled by graph family. Graph family includes: Barabási—Albert (BA), Erdős–Rényi (ER), Watts–Strogatz (WS), preferential attachment (PA), k-regular (Regular 3,4,6,10), star graph (Star) bipartite, and spatial random geometric graphs (Geo, Geosplit). (Left) Distribution of λ in the synthetic graph dataset used for training and validation. λ ranges from 0 to 1. (Right) Distribution of α in the synthetic graph dataset used for training and validation. α ranges from 0.81 to 1.95.

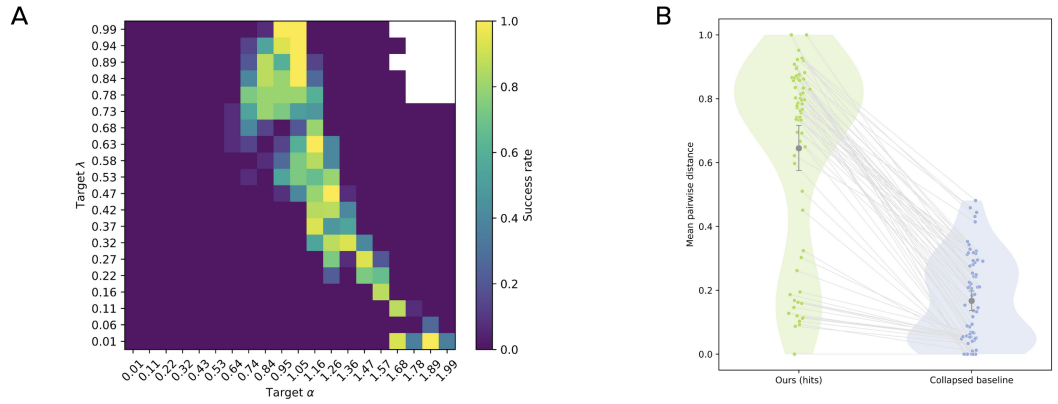

Figure 7: LGD model provides accurate and robust topological control. (A) Heatmap of success-within-tolerance rates across the (α, λ) grid. High success regions show that conditioning generalizes across both in-distribution and out-of-distribution graph families. (B) Mean pairwise 1-WL feature distance among graphs sampled at the same (α, λ) target. Our method (green) produces diverse solutions, while degree-preserving collapsed baselines (blue) show lower diversity. Lines connect paired runs.

Appendix

6 Model Architectures and Training

We trained a latent graph diffusion model using edge-enhanced graph transformers as the both the autoencoder, the regression head, and the denoising network (as described in [35]). The specific dimensions of the model is described below

Unsupervised structure encoder. We use a graph transformer structure encoder to encode graphs into the latent space. Node and edge attributes are first linearly embedded to 64-d features (LinearNode/LinearEdge encoders), and a virtual node/edge is added to facilitate global mixing [35]. The transformer has 3 layers with 4 attention heads per layer. Each layer uses a 64-d hidden size and produces a 32-d graph embedding at the output. The encoder and decoder are jointly trained to mini-

mize reconstruction loss of node and edge labels. During pretrainig of the autoencoder, conditioning labels at graph-level are not used. During joint training with the diffusion denoising network, the conditioning labels α and λ are embedded through a 2-layer MLP into a 64-d prefix that is additively fused to all node and edge embeddings with the token streams. We disable positional encodings for this process. We mask the node and edge labels with 30% probability during training. Attention uses full self-attention with edge enhancement and learned edge outputs (O_e, norm_e). Feed-forward edge updates are enabled, and a light MPNN block (ReLU, dropout 0.2) projects and updates edges between transformer blocks. We use batch normalization and residual connections. Dropout is 0.2 on activations and 0.5 on attention scores. We use mean pooling for tokens and additive pooling at the graph-level. The model is trained with L1 loss, dot-product edge decoding, batch size 140, AdamW (weight decay 10^{-5}), base lr 10^{-3}, cosine schedule with 50 warmup epochs, max 1000 epochs.

Regression encoder. We use the exact same backbone as the unsupervised structure encoder for predicting the conditioning labels (λ and α) from the graph space. The regression model is trained with L1 loss on the conditioning labels. The batch size is 140, base lr is 5×10^{-4}, and max 1000 epochs.

Conditional diffusion. We train a latent diffusion model with encoder output width 32 and denoiser (time-conditioned transformer) hidden size 64, 2 layers, 4 heads, temporal embedding 128, condition embedding 64, and cross-attention conditioning on $y = (\alpha^*, \lambda^*)$. We use AdamW (base lr 10^{-3}), cosine schedule with 50 warmup epochs, gradient-norm clipping, and batch size 84. Evaluation runs every 50 epochs with eval batch size 300; checkpoints are saved periodically and on best validation.

Losses. (i) **Unsupervised encoder:** masked node-label prediction and edge-existence prediction (no edge-label prediction), and graph-level reconstruction loss. (ii) **Regression encoder:** graph-level label regression loss only (accuracy of (α, λ) prediction via ℓ_1). (iii) **Diffusion:** latent noise-prediction objective $\mathcal{L}_{\text{diff}}$ as above; no label regression loss term.

Compute resources: For this study, each experiment was run on a single NVIDIA GeForce RTX 5090 (32GB RAM) and AMD Ryzen Threadripper 7960X 24-Core, 48-Thread CPU @ 4.802.90GHz.