Heterogeneous Coupled Diffusion for Graph Generation with α-Stable Node Feature Noise

Chenyu Tang[1] **Ercan Engin Kuruoglu**[1,†]

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[†]Correspondence to kuruoglu@sz.tsinghua.edu.cn

Abstract

Graph generation requires jointly modeling node semantics and graph structure. Existing coupled graph diffusion models mostly use light-tailed perturbations throughout the graph state, which may be too conservative for node features that encode rare semantic roles or long-tailed attribute patterns. We propose a heterogeneous coupled graph diffusion model that applies discrete-time symmetric α-stable noise to node features and Gaussian denoising diffusion probabilistic model (DDPM) noise to adjacency, while keeping reverse denoising joint over the full graph state. Using a Gaussian scale-mixture representation, we derive an exact augmented variational objective for the stable feature branch and a same-marginal coupled deterministic sampler. On a controlled benchmark, stable feature diffusion improves rare-role recall and joint generation quality, while Gaussian adjacency provides the more reliable structural bias. On molecular and generic graph benchmarks, keeping adjacency Gaussian and varying only the feature tail index often improves over both a retrained score-based graph diffusion baseline and a Gaussian-limit variant, though the best tail index remains dataset-dependent. These results support stable diffusion on features in coupled graph diffusion, while the case for Gaussian adjacency is primarily established by the controlled probe.

1. Introduction

Graph generation asks a model to sample both node attributes and the edge pattern that relates them. These two objects cannot usually be modeled independently: node features may encode semantic roles, while the validity of those roles depends on how nodes are connected. A useful graph generator therefore needs to preserve consistency between node-level semantics and graph-level structure.

Diffusion and score-based models are attractive for this setting because they learn to reverse a gradual noising process on the graph state. Recent graph diffusion methods instantiate this idea in several ways, including coupled continuous-state models, permutation-invariant continuous diffusion, and discrete denoising formulations (Ho et al., 2020; Song et al., 2021b,a; Jo et al., 2022; Huang et al., 2022; Vignac et al., 2023; Liu et al., 2025). We focus on the coupled node-edge line represented by Graph Diffusion via the System of Stochastic Differential Equations (GDSS), the score-based graph generator of Jo et al. (2022), where node and edge variables are denoised jointly from a shared noisy graph state rather than treated as weakly coupled components.

Existing coupled graph diffusion models mostly use light-tailed perturbations throughout this shared state. This uniform choice is stable, but it may not match the two graph components equally well. Node features can encode rare semantic roles or long-tailed attribute patterns, for which nonlocal

moves may help exploration beyond local Gaussian transport. By contrast, adjacency variables define the structural scaffold of the graph, where large jump-like updates can disrupt local repair. Heavy-tailed diffusion with α-stable Lévy noise provides a principled way to introduce nonlocal perturbations on continuous variables (Yoon et al., 2023; Shariatian et al., 2025), but it has not been integrated into coupled node-edge graph diffusion.

We propose a heterogeneous coupled diffusion model for graphs that uses discrete-time symmetric α-stable noise on node features and Gaussian denoising diffusion probabilistic model (DDPM) noise on adjacency, while keeping reverse denoising joint through a shared graph state. Using the Gaussian scale-mixture representation of stable noise (Shariatian et al., 2025), we derive an exact augmented variational objective for the feature branch together with a corresponding same-marginal coupled deterministic sampler in the DDPM / denoising diffusion implicit model (DDIM) spirit (Song et al., 2021a), giving a complete training-and-inference formulation for the proposed model. The resulting model is heterogeneous in its forward noising design but coupled in its reverse dynamics.

Our experiments support different parts of this division of labor at different levels. On a controlled benchmark that separates feature-side semantic transport from adjacency-side structural repair, stable feature diffusion improves rare-role recall and joint generation quality, while Gaussian adjacency is the more reliable bias for repair. On molecular and generic graph benchmarks, we keep adjacency Gaussian and vary only the feature tail index; under this fixed-structure setting, heavy-tailed feature diffusion often improves over both a GDSS baseline retrained from the official implementation under our experimental configuration and our Gaussian-limit variant, but no single tested tail index dominates every dataset: $\alpha = 1.7$ is strongest on QM9 and ego_small, whereas $\alpha = 1.5$ is preferred on several ZINC250k and community_small metrics.

Our contributions are threefold. First, we formulate a heterogeneous coupled graph diffusion model with stable node-feature perturbations and Gaussian adjacency perturbations under joint reverse denoising. Second, we derive an exact augmented variational objective for the stable feature branch together with its corresponding same-marginal coupled deterministic sampler, yielding a complete training-and-inference formulation for the model. Third, we show empirically that stable feature diffusion improves semantic transport on a controlled probe and often improves real-data generation quality when the adjacency branch is kept Gaussian; the stronger structure-side case for Gaussian adjacency is established primarily on the controlled probe.

2. Background

2.1. Diffusion, Score-Based, and Heavy-Tailed Foundations

Diffusion and score-based generative models provide the general backbone for our method; see Yang et al. (2024) for a broader survey. Denoising diffusion probabilistic models (DDPMs) formulate generation as learning the reverse of a discrete Markov noising chain (Ho et al., 2020), while score-based stochastic differential equation (SDE) models unify denoising score matching and diffusion in continuous time through reverse-time SDEs and probability-flow ordinary differential equations (ODEs) (Song et al., 2021b). Denoising diffusion implicit models (DDIMs) further show that the same training objective can also support non-Markovian deterministic same-marginal samplers (Song et al., 2021a). Our model inherits this DDPM/DDIM viewpoint, but adapts it to a coupled graph state with heterogeneous noising processes on node features and adjacency.

Within this broader diffusion literature, several works explore non-Gaussian and heavy-tailed perturbations. Stable laws provide a standard family for modeling heavy-tailed noise (Nolan, 2020). Denoising Diffusion Gamma Models and Heavy-Tailed Denoising Score Matching study alternatives to Gaussian noise (Nachmani et al., 2021; Deasy et al., 2021). Yoon et al. (2023) introduce isotropic α-stable Lévy processes into continuous-time score-based modeling and derive the corresponding reverse-time SDE and fractional denoising score matching objective. The Denoising Lévy

Probabilistic Models (DLPM) framework later revisits heavy-tailed diffusion from a DDPM perspective through a conditional-Gaussian scale-mixture construction, yielding explicit Gaussian backward densities conditioned on latent stable variables (Shariatian et al., 2025).

2.2. Graph Generation and Graph Diffusion

Graph representation learning provides a broad foundation for modeling graph-structured data (Hamilton, 2020), and deep graph generation has been surveyed across multiple generative families (Guo and Zhao, 2023). Before diffusion models, graph generation was studied through both autoregressive and non-autoregressive approaches. Autoregressive methods such as DeepGMG, GraphRNN, GraphAF, and GraphDF can capture detailed structural dependencies, but they rely on generation orderings or incur relatively high sampling cost (Li et al., 2018; You et al., 2018; Shi et al., 2020; Luo et al., 2021). Non-autoregressive alternatives, including GraphVAE, MoFlow, and GraphEBM, avoid explicit sequential decoding but still face trade-offs among expressiveness, tractable likelihoods, and graph validity (Simonovsky and Komodakis, 2018; Zang and Wang, 2020; Liu et al., 2021).

Score-based and diffusion models address these issues more directly. Niu et al. (2020) introduce an early score-based graph generator for adjacency matrices, while GDSS, the score-based graph generator of Jo et al. (2022), extends diffusion to coupled node-edge dynamics and emphasizes joint reverse modeling rather than weakly coupled denoising. Other graph diffusion lines broaden the overall landscape in different directions, including permutation-invariant continuous diffusion (GraphGDP), discrete denoising for categorical graphs (DiGress), and graph beta diffusion for mixed graph data (Huang et al., 2022; Vignac et al., 2023; Liu et al., 2025). We cite these works to place our method in the broader graph diffusion literature, but the methodological and empirical comparisons in this paper are centered on the coupled node-edge family most directly comparable to GDSS. Across these lines of work, a recurring challenge is to model node-edge dependence while choosing noise families that remain compatible with graph structure.

Our work sits specifically at the intersection of GDSS-style coupled graph diffusion and DLPM-style heavy-tailed diffusion. The main novelty is therefore not simply to apply stable noise to graph data. Rather, we address the within-family gap left open by prior work: how to incorporate the latent stable scale variables required by heavy-tailed DDPMs into a coupled node-edge reverse process. A naive combination of a DLPM feature diffuser with a GDSS-style coupled graph diffusion model would still leave unspecified the augmented graph state, the exact variational decomposition, and the conditioning structure that lets node and edge denoisers act on the same scale-aware noisy graph state. Our method fills this gap by applying DLPM-style scale augmentation only on the feature branch, keeping the adjacency branch Gaussian, and jointly parameterizing node and edge reverse updates from a shared augmented graph state. This yields a graph-aware heavy-tailed diffusion model rather than a simple plug-in replacement of Gaussian noise.

3. Method

We study general graph generation over the joint distribution $p_{\text{data}}(X, A)$, where $G = (X, A)$, $X \in \mathbb{R}^{N\times F}$ denotes node features, and $A \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix. Our model combines a heavy-tailed diffusion on node features (Yoon et al., 2023; Shariatian et al., 2025) with a Gaussian diffusion on adjacency, while keeping the reverse dynamics coupled through a shared graph state in the spirit of GDSS, the score-based graph generator of Jo et al. (2022). For compactness, notation used in the method and appendix proofs is collected in Table 4.

3.1. Problem Setup and Heterogeneous Forward Process

The forward process is defined component-wise as

$$q(G_{1:T} \mid G_0) = \prod_{t=1}^{T} q_X(X_t \mid X_{t-1})\, q_A(A_t \mid A_{t-1}), \qquad G_t = (X_t, A_t). \tag{1}$$

Feature branch. The node-feature branch follows a discrete-time symmetric α-stable diffusion (Nolan, 2020):

$$X_t = \gamma_t^X X_{t-1} + \sigma_t^X \xi_t^X, \qquad \xi_t^X \sim S(\alpha, 0, 0, 1), \tag{2}$$

where $\alpha \in (0, 2]$ is the tail index and $\{\gamma_t^X, \sigma_t^X\}_{t=1}^T$ is the feature schedule. Define

$$\bar{\gamma}_t^X = \prod_{i=1}^{t} \gamma_i^X, \qquad \bar{\sigma}_t^X = \left(\sum_{i=1}^{t} \left(\sigma_i^X \prod_{j=i+1}^{t} \gamma_j^X \right)^{\alpha} \right)^{1/\alpha}. \tag{3}$$

Then the feature marginal admits the closed form

$$X_t = \bar{\gamma}_t^X X_0 + \bar{\sigma}_t^X \epsilon^X, \qquad \epsilon^X \sim S(\alpha, 0, 0, 1). \tag{4}$$

Adjacency branch. The adjacency branch follows a standard Gaussian denoising diffusion probabilistic model (DDPM) (Ho et al., 2020):

$$A_t = \sqrt{1 - \beta_t^A}\, A_{t-1} + \sqrt{\beta_t^A}\, \epsilon_t^A, \qquad \epsilon_t^A \sim \mathcal{N}_{\text{sym}}(0, I), \tag{5}$$

where $\mathcal{N}_{\text{sym}}(0, I)$ denotes symmetric Gaussian noise compatible with an undirected adjacency matrix. Let

$$\alpha_t^A = 1 - \beta_t^A, \qquad \bar{\alpha}_t^A = \prod_{i=1}^{t} \alpha_i^A. \tag{6}$$

Then

$$A_t = \sqrt{\bar{\alpha}_t^A} A_0 + \sqrt{1 - \bar{\alpha}_t^A}\, \epsilon^A, \qquad \epsilon^A \sim \mathcal{N}_{\text{sym}}(0, I). \tag{7}$$

The induced noisy graph marginal at time t is

$$p_t(X_t, A_t) = \int p_{\text{data}}(X_0, A_0) q_X(X_t \mid X_0) q_A(A_t \mid A_0)\, dX_0\, dA_0. \tag{8}$$

As in coupled graph diffusion models such as GDSS, we do not write the reverse process as a product of independent reverse conditionals. Instead, we derive a joint reverse operator that updates node and edge states from the same noisy graph state.

3.2. Gaussian Scale-Mixture Representation and Scale-Aware Joint Denoising

The key step on the feature branch is the Gaussian scale-mixture representation of symmetric stable noise. Following the representation used in the Denoising Lévy Probabilistic Models (DLPM) framework (Shariatian et al., 2025), the stable perturbation is written in its sub-Gaussian form: for each noise level t, there exists a positive stable scale variable $\Lambda_t > 0$, whose law is determined by α, and a Gaussian latent $z_t^X \sim \mathcal{N}(0, I)$, independent of Λ_t, such that

$$\epsilon^X \overset{d}{=} \Lambda_t^{1/2} z_t^X. \tag{9}$$

Substituting this representation into Eq. (4) yields

$$X_t = \bar{\gamma}_t^X X_0 + \bar{\sigma}_t^X \Lambda_t^{1/2} z_t^X. \tag{10}$$

Conditioned on Λ_t, the feature branch becomes Gaussian:

$$q_X(X_t \mid X_0, \Lambda_t) = \mathcal{N}\left(\bar{\gamma}_t^X X_0, (\bar{\sigma}_t^X)^2 \Lambda_t I\right). \tag{11}$$

This conditional-Gaussian view makes the relevant noisy state explicit. For the exact augmented process used in variational training, the feature branch is Gaussian only after conditioning on the scale history up to time t. We therefore define the augmented graph state

$$\widetilde{G}_t = (X_t, A_t, \Lambda_{1:t}), \qquad \Lambda_{1:t} = (\Lambda_1, \ldots, \Lambda_t), \tag{12}$$

and write both training-time reverse factors and the deterministic reverse fields below as functions of $(X_t, A_t, \Lambda_{1:t}, t)$. The node and edge branches remain coupled because both are updated from the same scale-aware noisy graph state.

3.3. Variational Inference Objective

To obtain an exact variational formulation, we augment the feature branch with an independent scale path $\Lambda_{1:T} \sim q_\Lambda$, where $q_\Lambda(\Lambda_{1:T}) := \prod_{t=1}^T p_\Lambda(\Lambda_t)$ and p_Λ is the positive mixing law satisfying $\Lambda_t^{1/2} z_t^X \overset{d}{=} S(\alpha, 0, 0, 1)$ for $z_t^X \sim \mathcal{N}(0, I)$. Denoting the augmented path by $\widetilde{G}_{1:T} := (X_{1:T}, A_{1:T}, \Lambda_{1:T})$, the exact augmented forward process is

$$q(\widetilde{G}_{1:T} \mid G_0) = q_\Lambda(\Lambda_{1:T})\, q_X(X_{1:T} \mid X_0, \Lambda_{1:T})\, q_A(A_{1:T} \mid A_0), \tag{13}$$

where $q_X(X_{1:T} \mid X_0, \Lambda_{1:T}) = \prod_{t=1}^T \mathcal{N}\big(\gamma_t^X X_{t-1}, (\sigma_t^X)^2 \Lambda_t I\big)$ and $q_A(A_{1:T} \mid A_0) = \prod_{t=1}^T q_A(A_t \mid A_{t-1})$. Marginalizing $\Lambda_{1:T}$ recovers the original stable feature forward chain in Eq. (2), while conditioned on $\Lambda_{1:t}$ the feature posterior $q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t})$ is Gaussian in closed form.

For variational training, we match the scale-path prior to the forward law, $p(\Lambda_{1:T}) = q_\Lambda(\Lambda_{1:T})$, and define the training-time generative model

$$p_\Theta(G_0, \widetilde{G}_{1:T}) = p(\Lambda_{1:T})\, p_T^X(X_T \mid \Lambda_{1:T})\, p_T^A(A_T) \prod_{t=1}^T r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t})\, r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t}), \tag{14}$$

where $r_{\theta,t}^X$ and $r_{\phi,t}^A$ are Gaussian reverse families. The explicit terminal priors p_T^X and p_T^A, together with the exact pathwise derivation and the cancellation of the scale-path term, are given in Appendix B. These stochastic reverse factors are used for variational training, whereas the sampler below belongs to a different same-marginal deterministic family, exactly as in the DDPM/denoising diffusion implicit model (DDIM) relation (Ho et al., 2020; Song et al., 2021a).

The exact negative evidence lower bound (ELBO) of Eq. (14) is

$$\mathcal{L}_{\mathrm{VI}} := -\mathbb{E}_{q(\widetilde{G}_{1:T} \mid G_0)} \left[\log p_\Theta(G_0, \widetilde{G}_{1:T}) - \log q(\widetilde{G}_{1:T} \mid G_0)\right]. \tag{15}$$

With the matching choice $p(\Lambda_{1:T}) = q_\Lambda(\Lambda_{1:T})$, the scale-path contribution cancels exactly, and the exact negative ELBO separates into sampled $t \geq 2$ Kullback–Leibler (KL) terms plus terminal and reconstruction bookkeeping:

$$\mathcal{L}_{\mathrm{VI}} = \mathcal{L}_X^{\mathrm{VI}} + \mathcal{L}_A^{\mathrm{VI}} + C_X + C_A + \mathcal{R}_X + \mathcal{R}_A. \tag{16}$$

Here $\mathcal{L}_X^{\mathrm{VI}}$ and $\mathcal{L}_A^{\mathrm{VI}}$ are the sampled feature and adjacency KL terms in Eqs. (17) and (18); C_X, C_A are terminal constants for fixed priors, while $\mathcal{R}_X, \mathcal{R}_A$ are $t = 1$ reconstruction terms kept in the full bookkeeping rather than folded into the sampled KL objective. The full decomposition is given in Appendix B and Table 5.

For stochastic optimization, let $\pi(t)$ be a timestep sampling distribution with full support on $\{1,\dots,T\}$. Rewriting the timestep sums as expectations over $t \sim \pi$, the feature-side sampled KL term is

$$\mathcal{L}_X^{\mathrm{VI}} := \mathbb{E}_{t\sim\pi}\left[\frac{\mathbf{1}_{\{t\geq 2\}}}{\pi(t)}\mathbb{E}_q\left[D_{\mathrm{KL}}\big(q_X(X_{t-1}\mid X_t, X_0, \Lambda_{1:t}) \parallel r^X_{\theta,t}(X_{t-1}\mid X_t, A_t, \Lambda_{1:t})\big)\right]\right], \tag{17}$$

and the adjacency-side sampled KL term is

$$\mathcal{L}_A^{\mathrm{VI}} := \mathbb{E}_{t\sim\pi}\left[\frac{\mathbf{1}_{\{t\geq 2\}}}{\pi(t)}\mathbb{E}_q\left[D_{\mathrm{KL}}\big(q_A(A_{t-1}\mid A_t, A_0) \parallel r^A_{\phi,t}(A_{t-1}\mid X_t, A_t, \Lambda_{1:t})\big)\right]\right]. \tag{18}$$

Because both posteriors are Gaussian, the two KL terms admit closed-form expressions and can be estimated by sampling a single timestep $t \sim \pi$; the $1/\pi(t)$ factors make the estimator unbiased for the displayed $t \geq 2$ KL contributions, with the $t = 1$ terms recorded in Table 5. The deployed denoising loss is derived from, but is not a term-by-term optimization of, this exact ELBO: Appendix D shows how the sampled KL part yields an ELBO-derived same-target surrogate in the same-marginal sampler coordinates, which the implementation uses with positive proxy weights. Thus Eq. (15) defines the stochastic training-time reverse model, while the sampler below uses a DDPM/DDIM-style same-marginal deterministic family.

3.4. Coupled Deterministic Sampling

We next construct a deterministic reverse process from a same-marginal family. For the feature branch, let $\Lambda_{1:T} = (\Lambda_1, \dots, \Lambda_T)$ denote the time-indexed scale variables associated with the noise levels $1, \dots, T$. Conditioning on the pathwise scale sequence $\Lambda_{1:T}$ and a shared latent Gaussian z^X defines the non-Markovian family

$$\widetilde{q}_X(X_{1:T} \mid X_0, \Lambda_{1:T}, z^X) = \prod_{t=1}^{T} \delta\left(X_t - \bar{\gamma}_t^X X_0 - \bar{\sigma}_t^X \Lambda_t^{1/2} z^X\right). \tag{19}$$

Marginalizing over (Λ_t, z^X) at each fixed t recovers Eq. (4), so $\widetilde{q}_X$ has the same time marginals as the original stable forward process. Analogously, the adjacency branch admits the same-marginal family

$$\widetilde{q}_A(A_{1:T} \mid A_0, \epsilon^A) = \prod_{t=1}^{T} \delta\left(A_t - \sqrt{\bar{\alpha}_t^A} A_0 - \sqrt{1-\bar{\alpha}_t^A}\,\epsilon^A\right), \tag{20}$$

which shares the marginals of Eq. (7). The full proofs are deferred to Appendix C.

For deterministic sampling, we parameterize the coupled reverse fields by the scale-aware denoisers

$$\hat{z}_t^X = \mathcal{D}_\theta^X(X_t, A_t, \Lambda_{1:t}, t), \qquad \hat{\epsilon}_t^A = \mathcal{D}_\phi^A(X_t, A_t, \Lambda_{1:t}, t). \tag{21}$$

Given $\hat{z}_t^X$ and $\hat{\epsilon}_t^A$, we first reconstruct the clean graph estimates

$$\hat{X}_0 = \frac{X_t - \bar{\sigma}_t^X \Lambda_t^{1/2} \hat{z}_t^X}{\bar{\gamma}_t^X}, \qquad \hat{A}_0 = \frac{A_t - \sqrt{1-\bar{\alpha}_t^A}\,\hat{\epsilon}_t^A}{\sqrt{\bar{\alpha}_t^A}}. \tag{22}$$

For any earlier step $s < t$, the deterministic reverse updates are

$$X_s = \bar{\gamma}_s^X \hat{X}_0 + \bar{\sigma}_s^X \Lambda_s^{1/2} \hat{z}_t^X, \tag{23}$$

and

$$A_s = \sqrt{\bar{\alpha}_s^A}\,\hat{A}_0 + \sqrt{1-\bar{\alpha}_s^A}\,\hat{\epsilon}_t^A. \tag{24}$$

A derivation of Eqs. (22) to (24) from the same-marginal family is given in Appendix C.1. In the full reverse chain, we use $s = t - 1$. The joint reverse step is therefore

$$G_{t-1} = \Psi_\Theta(G_t, \Lambda_{1:t}, t), \qquad \Theta = \{\theta, \phi\}, \tag{25}$$

where the denoisers are conditioned on the current augmented noisy graph state $(X_t, A_t, \Lambda_{1:t})$, and the feature update to $t-1$ uses the corresponding pathwise scale entry Λ_{t-1}. This yields a coupled deterministic sampler that preserves the joint graph dependency structure while combining heavy-tailed feature transport with smooth adjacency denoising. Both branches remain continuous throughout diffusion and reverse denoising; task-specific discretization is applied only after the final sample G_0 is obtained.

4. Experiments

We first use a controlled diagnostic benchmark to isolate where heavy-tailed diffusion helps in the coupled graph model, and then evaluate the resulting feature-side design under fixed Gaussian adjacency on molecular and generic graph data.

4.1. Semantic Transport and Structural Repair Probe

We begin with a controlled synthetic benchmark designed to separate feature-side semantic transport from adjacency-side structural repair through role-structured graph constraints. Each graph contains 12 to 16 nodes with three roles: common, bridge, and rare_hub, with base probabilities 0.7, 0.2, and 0.1, respectively. Common nodes form local chain or clique components, bridge nodes must connect multiple components, and rare-hub nodes induce star-like structures without leaf-to-leaf edges. A final legality-preserving local rewiring step makes $A \mid X$ multimodal, so the benchmark isolates nonlocal semantic transport in X from local repair in A.

We compare four coupled variants that differ only in the noise family used on the feature and adjacency branches: Gaussian-X/Gaussian-A (GG), Stable-X/Gaussian-A (SG), Gaussian-X/Stable-A (GS), and Stable-X/Stable-A (SS). When a branch uses stable noise, we set $\alpha = 1.7$. All other architecture, optimization, and evaluation settings are fixed. We evaluate the models under two protocols. In the first, we sample full graphs from noise and report rare-role recall, joint error, and role-pair maximum mean discrepancy (MMD) (Gretton et al., 2012). In the second, we clamp X and sample only A, then report nearest-valid distance, repaired role-pair error, and repaired role-pair MMD. The fixed-X protocol acts as a negative control: once the feature assignment is fixed, the semantic transport problem is removed and the benchmark focuses on adjacency repair. Appendix Appendix E gives the full benchmark definition, including the fixed-X case-selection rule, the metric formulas, the candidate-based nearest-valid projector, and the legality-preserving rewiring algorithm.

Table 1 supports the intended division of labor between the two branches. In joint generation, switching the feature branch from Gaussian to stable improves performance within both adjacency families: SG improves over GG on rare recall, joint error, and role-pair MMD, and SS improves over GS on the same three metrics. At the same time, Gaussian-A is the more reliable structural bias. SG achieves the lowest joint error and role-pair MMD, while GG outperforms GS on both structural metrics.

The fixed-X probe sharpens this interpretation. Once the feature assignment is clamped, the stable-X advantage is no longer consistent across conditional adjacency metrics: SG improves the nearest-valid distance relative to GG, but GG remains best on repaired role-pair error and repaired role-pair MMD. Likewise, SS lowers repair distance relative to GS, but does not improve the other repaired statistics. Across both feature families, Gaussian-A yields the lower repair distance. These results indicate that heavy-tailed diffusion helps most on feature-side semantic transport, whereas Gaussian diffusion remains the safer choice for adjacency repair on this controlled benchmark.

Table 1. Controlled diagnostic benchmark. Best values are in bold.

Joint generation

Metric	GG	SG	GS	SS
Rare recall ↑	0.4380	0.6713	0.4173	**0.6926**
Joint error ↓	0.7550	**0.5362**	1.0856	0.6224
Role-pair MMD ↓	0.0523	**0.0249**	0.0580	0.0264

Fixed-X repair

Metric	GG	SG	GS	SS
Nearest-valid distance ↓	22.6185	**22.0990**	23.2799	22.5703
Repaired role-pair error ↓	**0.0527**	0.0608	0.0566	0.0627
Repaired role-pair MMD ↓	**0.0347**	0.0456	0.0394	0.0532

4.2. Experiments on Molecular Data and Generic Data

We next evaluate the feature-side tail index on two molecular datasets (QM9 and ZINC250k) (Ramakrishnan et al., 2014; Irwin et al., 2012) and two generic graph datasets (community_small and ego_small) following the benchmark setups used in prior graph-generation work (You et al., 2018). In this comparison, GDSS denotes the official implementation of the score-based graph generator of Jo et al. (2022), retrained under our experimental configuration and evaluated with the same metric pipeline used for our models; it is therefore not the published numbers reported in Jo et al. (2022). We report this retrained GDSS baseline alongside our own tail-index sweep at $\alpha \in \{2.0, 1.7, 1.5\}$. The $\alpha = 2.0$ row is our own Gaussian-limit variant run in the same sweep and is therefore reported separately from GDSS. For molecular data, we report validity without correction, uniqueness at 10,000 samples, novelty, Fréchet ChemNet Distance (FCD), and neighborhood subgraph pairwise distance kernel maximum mean discrepancy (NSPDK MMD) (Preuer et al., 2018; Costa and De Grave, 2010; Gretton et al., 2012). For generic graph data, we report degree, clustering, orbit, spectral, and mean MMD (Gretton et al., 2012); in particular, spectral and mean MMD are obtained by re-evaluating the retrained GDSS checkpoints under the same common pipeline, even though these metrics are not listed in the original GDSS paper. These real-data experiments are deliberately feature-side ablations: the adjacency branch is kept Gaussian throughout, and only the feature tail index is varied. They therefore test whether heavy-tailed feature diffusion helps under fixed Gaussian adjacency, not whether Gaussian adjacency outperforms stable adjacency on these datasets. As a bounded check, Table 7 reports QM9-only stable-adjacency variants; these checks did not improve over the reported Stable-X/Gaussian-A setting except on novelty for the dual-stable variant, so we treat broader stable-adjacency evaluation as future work rather than as a main real-data claim.

On molecular data, the moderate heavy-tailed setting $\alpha = 1.7$ is the clearest improvement on QM9, where it is best on all five metrics in Table 2. ZINC250k shows a less uniform pattern: $\alpha = 1.7$ gives the highest validity and uniqueness, while $\alpha = 1.5$ is best on FCD and NSPDK MMD. The separate GDSS and $\alpha = 2.0$ rows make clear that the retrained official GDSS baseline and our own Gaussian-limit variant should not be conflated. This indicates that heavy-tailed feature diffusion improves molecular generation quality, while the best tail index for distributional fidelity remains dataset-dependent.

Table 2. Results on molecular datasets. GDSS denotes the retrained score-based graph diffusion baseline of Jo et al. (2022) evaluated with our common pipeline. Best values are in bold.

	Validity w/o correction ↑	Unique@10000 ↑	Novelty ↑	FCD ↓	NSPDK MMD ↓
QM9					
GDSS	0.9496	0.9703	0.819787	2.894014	0.0050368
$\alpha = 2.0$	0.9352	0.9645	0.839794	2.835053	0.004969
$\alpha = 1.7$	**0.9726**	**0.9899**	**0.862251**	**2.778251**	**0.004516**
$\alpha = 1.5$	0.9629	0.9829	0.861939	3.059429	0.005791
ZINC250k					
GDSS	0.9387	0.9884	**1.0**	17.438318	0.054645
$\alpha = 2.0$	0.9248	0.9923	**1.0**	17.884385	0.058262
$\alpha = 1.7$	**0.9688**	**1.0**	**1.0**	18.301	0.063283
$\alpha = 1.5$	0.9121	0.9997	**1.0**	**17.140461**	**0.052196**

The generic graph results show the same broad trend, but again with dataset-specific optima. On ego_small, $\alpha = 1.7$ is best on four of the five reported metrics. On community_small, $\alpha = 1.5$ is best on degree, clustering, and mean MMD, while GDSS remains best on orbit MMD and $\alpha = 1.7$ slightly improves spectral MMD. Taken together with the molecular results, these experiments show that heavy-tailed feature diffusion often improves over both the retrained GDSS baseline and our $\alpha = 2.0$ Gaussian-limit variant, but they do not identify a universal best tail index. In this sweep, $\alpha = 1.7$ is strongest on QM9 and ego_small, whereas $\alpha = 1.5$ is preferred for FCD and NSPDK on ZINC250k and for degree, clustering, and mean MMD on community_small.

Table 3. Results on generic graph datasets. GDSS denotes the retrained score-based graph diffusion baseline of Jo et al. (2022) evaluated with our common pipeline. Best values are in bold.

	Degree MMD ↓	Cluster MMD ↓	Orbit MMD ↓	Spectral MMD ↓	Mean MMD ↓
community_small					
GDSS	0.066815	0.070934	**0.007328**	0.191255	0.084083
$\alpha = 2.0$	0.068424	0.072629	0.007943	0.190651	0.082582
$\alpha = 1.7$	0.041738	0.074076	0.012997	**0.189788**	0.079650
$\alpha = 1.5$	**0.033747**	**0.045803**	0.011432	0.189947	**0.070232**
ego_small					
GDSS	0.040430	0.056811	0.003784	**0.050736**	0.040291
$\alpha = 2.0$	0.042055	0.059675	0.004392	0.053433	0.039716
$\alpha = 1.7$	**0.021312**	**0.046981**	**0.003751**	0.050789	**0.030708**
$\alpha = 1.5$	0.043736	0.060329	0.004999	0.052100	0.037940

5. Conclusion

We introduced a heterogeneous coupled diffusion model for graph generation that assigns different noise families to different graph components while preserving joint reverse denoising. Concretely, node features are diffused with a discrete-time symmetric α-stable process, adjacency with a Gaussian DDPM process, and both branches are denoised from a shared augmented graph state. Using the Gaussian scale-mixture representation of stable noise, we derived an exact augmented variational objective for the feature branch together with its corresponding same-marginal coupled deterministic sampler in the DDPM/DDIM spirit. This gives a complete training-and-inference formulation showing that heterogeneous forward noising can be incorporated into graph diffusion without sacrificing the node-edge coupling that is central to coherent graph generation.

Empirically, the evidence is strongest for the feature-side part of this design, with a separate controlled probe supporting the structure-side claim. On the controlled probe, heavy-tailed feature

diffusion improves rare-role recovery and joint generation quality, while Gaussian adjacency is the more reliable inductive bias for repair. On molecular and generic graph benchmarks, we keep adjacency Gaussian and vary only the feature tail index; under this fixed-structure setting, heavy-tailed feature diffusion often improves over both a GDSS baseline retrained from the official implementation under our experimental configuration and our Gaussian-limit variant, but no single tested tail index dominates every dataset: $\alpha = 1.7$ is strongest on QM9 and ego_small, whereas $\alpha = 1.5$ is preferred on several ZINC250k and community_small metrics. Taken together, these results suggest that heavier-tailed feature diffusion can benefit coupled graph generation in this setting, while the stronger case for Gaussian adjacency as the preferred structural noise family is currently supported mainly by the controlled probe.

More broadly, our results suggest that heterogeneous noising can be combined with a shared reverse process without sacrificing coupled graph generation. Within this setup, stable feature transport appears to broaden semantic exploration, while Gaussian adjacency is the safer adjacency choice on the controlled probe and remains better-performing than the compute-limited QM9 stable-adjacency checks except on novelty. An important next step is to test adjacency-noise choices systematically across real datasets beyond this QM9 check, alongside making the tail index and the allocation of noise families adaptive across datasets, graph substructures, and timesteps, and extending the framework to larger, more discrete, and more strongly constrained graph domains.

References

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 27th International Conference on Machine Learning*, pages 255–262, 2010. URL https://icml.cc/Conferences/2010/papers/347.pdf.

Jacob Deasy, Nikola Simidjievski, and Pietro Liò. Heavy-tailed denoising score matching. *arXiv preprint arXiv:2112.09788*, 2021. doi: 10.48550/arXiv.2112.09788. URL https://arxiv.org/abs/2112.09788.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL https://jmlr.org/papers/v13/gretton12a.html.

Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5370–5390, 2023. doi: 10.1109/TPAMI.2022.3214832.

William L. Hamilton. *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-031-01588-5.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. GraphGDP: Generative diffusion processes for permutation invariant graph generation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 201–210. IEEE, 2022. doi: 10.1109/ICDM54844.2022.00030.

John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52 (7):1757–1768, 2012. doi: 10.1021/ci3001277.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10362–10383. PMLR, 2022. URL https://proceedings.mlr.press/v162/jo22a.html.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018. doi: 10.48550/arXiv.1803.03324. URL https://arxiv.org/abs/1803.03324.

Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. GraphEBM: Molecular graph generation with energy-based models. In *Energy-Based Models Workshop at ICLR*, 2021. URL https://openreview.net/forum?id=Gc51PtL_zYw.

Xinyang Liu, Yilin He, Bo Chen, and Mingyuan Zhou. Advancing graph generation through beta diffusion. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=x1An5a3U9I.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. GraphDF: A discrete flow model for molecular graph generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7192–7203. PMLR, 2021. URL https://proceedings.mlr.press/v139/luo21a.html.

Eliya Nachmani, Robin San Roman, and Lior Wolf. Denoising diffusion gamma models. *arXiv preprint arXiv:2110.05948*, 2021. doi: 10.48550/arXiv.2110.05948. URL https://arxiv.org/abs/2110.05948.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4474–4484. PMLR, 2020. URL https://proceedings.mlr.press/v108/niu20a.html.

John P. Nolan. *Univariate Stable Distributions: Models for Heavy Tailed Data*. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-52915-4.

Kristian Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet ChemNet Distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018. doi: 10.1021/acs.jcim.8b00234.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014. doi: 10.1038/sdata.2014.22.

Dario Shariatian, Umut Simsekli, and Alain Oliviero Durmus. Heavy-tailed diffusion with denoising Lévy probabilistic models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=SYmUS6qRub.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: A flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1esMkHYPr.

Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, Lecture Notes in Computer Science, pages 412–422. Springer International Publishing, 2018. doi: 10.1007/978-3-030-01418-6_41.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

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)*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

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)*, 2023. URL https://openreview.net/forum?id=UaAD-Nu86WX.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2024. doi: 10.1145/3626235. Article 105.

Eunbi Yoon, Keehun Park, Sungwoong Kim, and Sungbin Lim. Score-based generative models with Lévy processes. In *Advances in Neural Information Processing Systems*, volume 36, pages 40694–40707. Curran Associates, Inc., 2023. doi: 10.52202/075280-1772. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8011b23e1dc3f57e1b6211ccad498919-Paper-Conference.pdf.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5708–5717. PMLR, 2018. URL https://proceedings.mlr.press/v80/you18a.html.

Chence Zang and Fei Wang. MoFlow: An invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 617–626. ACM, 2020. doi: 10.1145/3394486.3403104.

A. Notation for Method and Appendix Proofs

Table 4: Notation used in the method and appendix proofs.

Symbol(s)	**Meaning**
$G = (X, A)$, $G_t = (X_t, A_t)$	Clean graph and noisy graph at timestep t.
$X \in \mathbb{R}^{N\times F}$, $A \in \mathbb{R}^{N\times N}$	Node-feature matrix with N nodes and F features, and adjacency matrix.
α	Tail index of the symmetric stable feature noise; $\alpha = 2$ is the Gaussian limit.
$\gamma_t^X, \sigma_t^X, \bar{\gamma}_t^X, \bar{\sigma}_t^X$	Feature-branch step coefficients and cumulative stable marginal coefficients.
$\beta_t^A, \alpha_t^A, \bar{\alpha}_t^A$	Adjacency-branch denoising diffusion probabilistic model (DDPM) schedule, with $\alpha_t^A = 1 - \beta_t^A$.
ξ_t^X, ϵ^X	Symmetric stable feature-noise variables in the stepwise and marginal processes.
z_t^X, z^X	Gaussian latent variables used by the scale-mixture and same-marginal feature representations.
ϵ_t^A, ϵ^A	Symmetric Gaussian adjacency noise variables.
$\Lambda_t, \Lambda_{1:t}$	Positive stable scale variable at timestep t and its history.
$\widetilde{G}_t = (X_t, A_t, \Lambda_{1:t})$	Scale-augmented noisy graph state used by the variational formulation.
$r_{\theta,t}^X, r_{\phi,t}^A$	Stochastic reverse families for feature and adjacency training factors.
$\mathcal{D}_\theta^X, \mathcal{D}_\phi^A$	Scale-aware denoisers used by the deterministic sampler.
$\mathcal{L}_{\mathrm{VI}}, \mathcal{L}_X^{\mathrm{VI}}, \mathcal{L}_A^{\mathrm{VI}}$	Exact negative evidence lower bound (ELBO) and its sampled feature/adjacency Kullback–Leibler (KL) components.
$C_X, C_A, \mathcal{R}_X, \mathcal{R}_A$	Terminal and reconstruction bookkeeping terms in the full ELBO.
$\pi(t)$	Timestep sampling distribution used for stochastic optimization.
$\widetilde{G}_{1:T}$	Augmented path $(X_{1:T}, A_{1:T}, \Lambda_{1:T})$.
q_Λ, p_Λ	Forward scale-path law and one-step stable mixing law.
p_T^X, p_T^A	Terminal priors for the feature and adjacency branches.
$v_t^X(\Lambda_{1:t})$, η_t^X	Pathwise feature standard deviation and Gaussian feature noise in the exact augmented Markov chain.
$\widetilde{\mu}_t^X, \widetilde{\nu}_t^X$, $\widetilde{\mu}_t^A, \widetilde{\beta}_t^A$	Exact posterior means and variances for the feature and adjacency reverse conditionals.
$U_t = (X_t, A_t, \Lambda_{1:t}, t)$	Common conditioning variable for the scale-aware denoising heads.
$g_{\theta,t}^X, g_{\phi,t}^A$	Training-time denoising heads in the exact pathwise Gaussian coordinates.
ρ_t^X, ρ_t^A	Fixed reverse variances used by the variational reverse families.
ω_t^X, ω_t^A	Exact denoising weights obtained from the Gaussian Kullback–Leibler (KL) reductions.
$\zeta_t^X, c_t^X, h_{\theta,t}^X$	Same-marginal feature target, coordinate-change scale factor, and practical feature predictor.
λ_t^X, λ_t^A	Positive deployed proxy weights for feature and adjacency denoising.
$\hat{z}_t^X, \hat{\epsilon}_t^A, \hat{X}_0, \hat{A}_0$	Network-predicted latent variables and reconstructed clean graph estimates used by the sampler.
$\widehat{X}_0^{\mathrm{oracle}}, \widehat{A}_0^{\mathrm{oracle}}$	Clean-state estimates obtained in the sampler derivation when oracle latent variables are known.

B. Exact ELBO Decomposition of the Augmented Heterogeneous Diffusion

For a fixed clean graph $G_0 = (X_0, A_0)$, recall the exact augmented forward path

$$q(\widetilde{G}_{1:T} \mid G_0) = q_\Lambda(\Lambda_{1:T})\, q_X(X_{1:T} \mid X_0, \Lambda_{1:T})\, q_A(A_{1:T} \mid A_0), \tag{26}$$

with

$$q_X(X_{1:T} \mid X_0, \Lambda_{1:T}) = \prod_{t=1}^{T} q_X(X_t \mid X_{t-1}, \Lambda_t), \qquad q_A(A_{1:T} \mid A_0) = \prod_{t=1}^{T} q_A(A_t \mid A_{t-1}), \tag{27}$$

and the training-time generative model

$$p_\Theta(G_0, \widetilde{G}_{1:T}) = p(\Lambda_{1:T})\, p_T^X(X_T \mid \Lambda_{1:T})\, p_T^A(A_T) \prod_{t=1}^{T} r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t}) r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t}). \tag{28}$$

In this appendix, these terminal priors are taken to be fixed variance-matching Gaussians. Define the terminal feature variance

$$\Sigma_T^X(\Lambda_{1:T}) := \left(\sum_{i=1}^{T} \left(\sigma_i^X \prod_{j=i+1}^{T} \gamma_j^X \right)^2 \Lambda_i \right) I. \tag{29}$$

We then choose

$$p_T^X(X_T \mid \Lambda_{1:T}) := \mathcal{N}\big(0, \Sigma_T^X(\Lambda_{1:T})\big), \qquad p_T^A(A_T) := \mathcal{N}_{\mathrm{sym}}\big(0, (1 - \bar{\alpha}_T^A) I\big). \tag{30}$$

Thus the adjacency terminal prior matches the exact finite-T forward terminal covariance, while the feature terminal prior matches the exact conditional terminal covariance of the Gaussianized chain. If $\bar{\gamma}_T^X = 0$ and $\bar{\alpha}_T^A = 0$, these priors coincide with the corresponding forward terminal laws; otherwise the residual mismatch appears exactly in the terminal KL terms C_X and C_A below. The negative ELBO conditioned on G_0 is

$$\mathcal{L}_{\mathrm{VI}}(G_0) := \mathbb{E}_{q(\widetilde{G}_{1:T} \mid G_0)} \left[\log \frac{q(\widetilde{G}_{1:T} \mid G_0)}{p_\Theta(G_0, \widetilde{G}_{1:T})} \right]. \tag{31}$$

Averaging Eq. (31) over $G_0 \sim p_{\mathrm{data}}$ gives the full training objective.

Step 1: Expand the log-ratio. Using Eqs. (26) and (28), we obtain

$$\begin{aligned} \mathcal{L}_{\mathrm{VI}}(G_0) = \mathbb{E}_q \Big[& \underbrace{\log q_\Lambda(\Lambda_{1:T}) - \log p(\Lambda_{1:T})}_{\text{scale-path term}} \\ & + \log q_X(X_{1:T} \mid X_0, \Lambda_{1:T}) - \log p_T^X(X_T \mid \Lambda_{1:T}) - \sum_{t=1}^{T} \log r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t}) \\ & + \log q_A(A_{1:T} \mid A_0) - \log p_T^A(A_T) - \sum_{t=1}^{T} \log r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t}) \Big], \end{aligned} \tag{32}$$

where $\mathbb{E}_q[\cdot]$ is shorthand for expectation under $q(\widetilde{G}_{1:T} \mid G_0)$.

Step 2: Exact cancellation of the scale-path term. By construction, the prior over the scale path is chosen to match the forward scale law,

$$p(\Lambda_{1:T}) = q_\Lambda(\Lambda_{1:T}) = \prod_{t=1}^{T} p_\Lambda(\Lambda_t). \tag{33}$$

Therefore the scale-path contribution cancels *pointwise*:

$$\log q_\Lambda(\Lambda_{1:T}) - \log p(\Lambda_{1:T}) = 0 \qquad \text{for every } \Lambda_{1:T}, \tag{34}$$

and hence

$$\mathbb{E}_q\big[\log q_\Lambda(\Lambda_{1:T}) - \log p(\Lambda_{1:T})\big] = 0. \tag{35}$$

After this cancellation,

$$\mathcal{L}_{\mathrm{VI}}(G_0) = \mathcal{J}_X(G_0) + \mathcal{J}_A(G_0), \tag{36}$$

where

$$\mathcal{J}_X(G_0) := \mathbb{E}_q\left[\log q_X(X_{1:T} \mid X_0, \Lambda_{1:T}) - \log p_T^X(X_T \mid \Lambda_{1:T}) - \sum_{t=1}^{T} \log r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t})\right], \tag{37}$$

$$\mathcal{J}_A(G_0) := \mathbb{E}_q\left[\log q_A(A_{1:T} \mid A_0) - \log p_T^A(A_T) - \sum_{t=1}^{T} \log r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t})\right]. \tag{38}$$

The split in Eq. (36) is exact because, conditioned on $(G_0, \Lambda_{1:T})$, the forward path factorizes into a feature chain and an adjacency chain, so the log-density separates additively into an X-part and an A-part.

Step 3: Reverse factorization of the two forward chains. Since the feature branch is a Markov chain conditioned on $\Lambda_{1:T}$,

$$q_X(X_{1:T} \mid X_0, \Lambda_{1:T}) = q_X(X_T \mid X_0, \Lambda_{1:T}) \prod_{t=2}^{T} q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t}), \tag{39}$$

where future scales $\Lambda_{t+1:T}$ are irrelevant once $(X_t, X_0, \Lambda_{1:t})$ are given. Likewise, the Gaussian adjacency branch satisfies

$$q_A(A_{1:T} \mid A_0) = q_A(A_T \mid A_0) \prod_{t=2}^{T} q_A(A_{t-1} \mid A_t, A_0). \tag{40}$$

Equivalently, the exact posterior over the previous state factorizes as

$$q(X_{t-1}, A_{t-1} \mid X_t, A_t, G_0, \Lambda_{1:t}) = q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t}) q_A(A_{t-1} \mid A_t, A_0). \tag{41}$$

Substituting Eqs. (39) and (40) into Eqs. (37) and (38) gives

$$\mathcal{J}_X(G_0) = \mathbb{E}_q\left[\log \frac{q_X(X_T \mid X_0, \Lambda_{1:T})}{p_T^X(X_T \mid \Lambda_{1:T})} + \sum_{t=2}^{T} \log \frac{q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t})}{r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t})} - \log r_{\theta,1}^X(X_0 \mid X_1, A_1, \Lambda_1)\right], \tag{42}$$

$$\mathcal{J}_A(G_0) = \mathbb{E}_q\left[\log \frac{q_A(A_T \mid A_0)}{p_T^A(A_T)} + \sum_{t=2}^{T} \log \frac{q_A(A_{t-1} \mid A_t, A_0)}{r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t})} - \log r_{\phi,1}^A(A_0 \mid X_1, A_1, \Lambda_1)\right]. \tag{43}$$

Step 4: Convert the log-ratio terms into KLs. Taking expectation over the corresponding exact posterior in each ratio yields the standard KL form:

$$\begin{aligned}\mathcal{J}_X(G_0) = &\underbrace{\mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_X(X_T \mid X_0, \Lambda_{1:T}) \,\|\, p_T^X(X_T \mid \Lambda_{1:T})\big)\big]}_{=:\, C_X(G_0)} \\ &+ \sum_{t=2}^{T} \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t}) \,\|\, r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t})\big)\big] \\ &+ \underbrace{\mathbb{E}_q\big[-\log r_{\theta,1}^X(X_0 \mid X_1, A_1, \Lambda_1)\big]}_{=:\, \mathcal{R}_X(G_0)}, \end{aligned} \tag{44}$$

$$\begin{aligned}\mathcal{J}_A(G_0) = &\underbrace{\mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_A(A_T \mid A_0) \,\|\, p_T^A(A_T)\big)\big]}_{=:\, C_A(G_0)} \\ &+ \sum_{t=2}^{T} \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_A(A_{t-1} \mid A_t, A_0) \,\|\, r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t})\big)\big] \\ &+ \underbrace{\mathbb{E}_q\big[-\log r_{\phi,1}^A(A_0 \mid X_1, A_1, \Lambda_1)\big]}_{=:\, \mathcal{R}_A(G_0)}. \end{aligned} \tag{45}$$

Combining Eqs. (36), (44) and (45) yields the exact negative ELBO decomposition

$$\begin{aligned}\mathcal{L}_{\mathrm{VI}}(G_0) = &\sum_{t=2}^{T} \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t}) \,\|\, r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t})\big)\big] \\ &+ \sum_{t=2}^{T} \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_A(A_{t-1} \mid A_t, A_0) \,\|\, r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t})\big)\big] \\ &+ \mathcal{R}_X(G_0) + \mathcal{R}_A(G_0) + C_X(G_0) + C_A(G_0). \end{aligned} \tag{46}$$

Averaging Eq. (46) over $G_0 \sim p_{\mathrm{data}}$ gives the full objective.

Relation to the main-text shorthand. The decomposition in Eq. (46) is the exact appendix-level decomposition. It contains four kinds of terms: the feature-side KL sum over $t \geq 2$, the adjacency-side KL sum over $t \geq 2$, the terminal terms C_X and C_A, and the reconstruction terms $\mathcal{R}_X$ and $\mathcal{R}_A$ coming from the $t = 1$ reverse factors. The main text displays the sampled $t \geq 2$ KL contributions explicitly in Eqs. (17) and (18) and lists the terminal and reconstruction terms separately in Eq. (16). If we isolate the two displayed main-text quantities as

$$\overline{\mathcal{L}}_X^{\mathrm{VI}}(G_0) := \sum_{t=2}^{T} \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t}) \,\|\, r_{\theta,t}^X(X_{t-1} \mid X_t, A_t, \Lambda_{1:t})\big)\big], \tag{47}$$

and

$$\overline{\mathcal{L}}_A^{\mathrm{VI}}(G_0) := \sum_{t=2}^{T} \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_A(A_{t-1} \mid A_t, A_0) \,\|\, r_{\phi,t}^A(A_{t-1} \mid X_t, A_t, \Lambda_{1:t})\big)\big], \tag{48}$$

then the exact ELBO takes the form

$$\mathcal{L}_{\mathrm{VI}}(G_0) = \overline{\mathcal{L}}_X^{\mathrm{VI}}(G_0) + \overline{\mathcal{L}}_A^{\mathrm{VI}}(G_0) + C_X(G_0) + C_A(G_0) + \mathcal{R}_X(G_0) + \mathcal{R}_A(G_0). \tag{49}$$

Which terms are true constants? The terms C_X and C_A are parameter-independent provided that the terminal priors p_T^X and p_T^A are fixed a priori and the forward schedule is fixed. Indeed, they only compare the exact forward terminal marginals against fixed terminal priors and do not involve (θ, ϕ). In contrast, the reconstruction terms $\mathcal{R}_X$ and $\mathcal{R}_A$ are *not* constants unless $r_{\theta,1}^X$ and $r_{\phi,1}^A$ are fixed or handled outside the trainable reverse family.

Table 5. Bookkeeping of the exact negative ELBO.

Term	**Expression**	**Origin and parameter dependence**
Scale-path term	$\mathbb{E}_q[\log q_\Lambda(\Lambda_{1:T}) - \log p(\Lambda_{1:T})] = 0$	Exact cancellation because $p(\Lambda_{1:T}) = q_\Lambda(\Lambda_{1:T})$ pointwise; parameter-free.
Feature KLs	$\sum_{t=2}^{T} \mathbb{E}_q[D_{\mathrm{KL}}(q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t}) \| r^X_{\theta,t}(\cdot \mid X_t, A_t, \Lambda_{1:t}))]$	Reverse factorization of the exact feature chain; depends on θ.
Adjacency KLs	$\sum_{t=2}^{T} \mathbb{E}_q[D_{\mathrm{KL}}(q_A(A_{t-1} \mid A_t, A_0) \| r^A_{\phi,t}(\cdot \mid X_t, A_t, \Lambda_{1:t}))]$	Reverse factorization of the exact adjacency chain; depends on ϕ.
Feature reconstruction	$\mathcal{R}_X = -\mathbb{E}_q \log r^X_{\theta,1}(X_0 \mid X_1, A_1, \Lambda_1)$	The $t = 1$ reverse factor; parameter-dependent unless fixed.
Adjacency reconstruction	$\mathcal{R}_A = -\mathbb{E}_q \log r^A_{\phi,1}(A_0 \mid X_1, A_1, \Lambda_1)$	The $t = 1$ reverse factor; parameter-dependent unless fixed.
Feature terminal constant	$C_X = \mathbb{E}_q[D_{\mathrm{KL}}(q_X(X_T \mid X_0, \Lambda_{1:T}) \| p^X_T(X_T \mid \Lambda_{1:T}))]$	Terminal prior mismatch on the feature branch; parameter-free if p^X_T is fixed.
Adjacency terminal constant	$C_A = \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q_A(A_T \mid A_0) \| p^A_T(A_T)\big)\big]$	Terminal prior mismatch on the adjacency branch; parameter-free if p^A_T is fixed.

Unbiased single-timestep estimator. For any timestep sampling law π with full support on $\{1, \dots, T\}$,

$$\sum_{t=2}^{T} \mathbb{E}_q[f_t] = \mathbb{E}_{t\sim\pi}\left[\frac{\mathbf{1}_{\{t\geq 2\}}}{\pi(t)}\, \mathbb{E}_q[f_t]\right], \tag{50}$$

so the feature and adjacency KL sums in Eq. (46) admit unbiased single-timestep estimators. The reconstruction terms $\mathcal{R}_X$ and $\mathcal{R}_A$, when retained, must be handled separately. This is the precise sense in which the main-text stochastic objective estimates the displayed $t \geq 2$ KL contributions.

C. Proofs for the Same-Marginal Families

For a path law $q_\Lambda(\Lambda_{1:T})$ on $\Lambda_{1:T} = (\Lambda_1, \dots, \Lambda_T)$, let

$$q_{\Lambda,t}(\lambda) := \int q_\Lambda(\Lambda_{1:T})\, d\Lambda_{1:t-1}\, d\Lambda_{t+1:T} \tag{51}$$

denote the time-t marginal of the scale path. In our construction, $q_\Lambda(\Lambda_{1:T}) = \prod_{s=1}^{T} p_\Lambda(\Lambda_s)$, hence $q_{\Lambda,t} = p_\Lambda$ for every t. We nevertheless state the results below in terms of the pathwise law $q_\Lambda(\Lambda_{1:T})$ and its marginals $q_{\Lambda,t}$.

Proposition 1 (Feature-branch same-marginal property). *Let $z^X \sim \mathcal{N}(0, I)$ be independent of $\Lambda_{1:T} \sim q_\Lambda(\Lambda_{1:T})$, and assume that for every $t \in \{1, \dots, T\}$ the time-t scale marginal satisfies $q_{\Lambda,t} = p_\Lambda$, where p_Λ is the mixing law such that $\Lambda_t^{1/2} z^X \overset{d}{=} \epsilon^X$ with $\epsilon^X \sim S(\alpha, 0, 0, 1)$. Define*

$$\widetilde{q}_X(X_{1:T} \mid X_0, \Lambda_{1:T}, z^X) = \prod_{s=1}^{T} \delta\Big(X_s - \bar{\gamma}^X_s X_0 - \bar{\sigma}^X_s \Lambda_s^{1/2} z^X\Big). \tag{52}$$

Then for every $t \in \{1, \dots, T\}$,

$$\widetilde{q}_X(X_t \mid X_0) := \int \widetilde{q}_X(X_{1:T} \mid X_0, \Lambda_{1:T}, z^X) q_\Lambda(\Lambda_{1:T}) \mathcal{N}(z^X; 0, I) \, dX_{1:t-1} \, dX_{t+1:T} \, d\Lambda_{1:T} \, dz^X = q_X(X_t \mid X_0), \tag{53}$$

where $q_X(X_t \mid X_0)$ is the stable marginal in Eq. (4).

Proof. Fix t. Integrating out X_s for $s \neq t$ against the Dirac factors gives

$$\widetilde{q}_X(X_t \mid X_0) = \int \delta\Big(X_t - \bar{\gamma}_t^X X_0 - \bar{\sigma}_t^X \Lambda_t^{1/2} z^X\Big) \, q_\Lambda(\Lambda_{1:T}) \mathcal{N}(z^X; 0, I) \, d\Lambda_{1:T} \, dz^X \tag{54}$$

$$= \int \delta\Big(X_t - \bar{\gamma}_t^X X_0 - \bar{\sigma}_t^X \lambda^{1/2} z^X\Big) \, q_{\Lambda,t}(\lambda) \mathcal{N}(z^X; 0, I) \, d\lambda \, dz^X, \tag{55}$$

where the second line uses the definition of the time-t marginal $q_{\Lambda,t}$. By the assumption $q_{\Lambda,t} = p_\Lambda$, the right-hand side is exactly the law of $\bar{\gamma}_t^X X_0 + \bar{\sigma}_t^X \Lambda_t^{1/2} z^X$. By the defining sub-Gaussian representation of the stable noise, $\Lambda_t^{1/2} z^X \overset{d}{=} \epsilon^X$ with $\epsilon^X \sim S(\alpha, 0, 0, 1)$. Hence

$$\bar{\gamma}_t^X X_0 + \bar{\sigma}_t^X \Lambda_t^{1/2} z^X \overset{d}{=} \bar{\gamma}_t^X X_0 + \bar{\sigma}_t^X \epsilon^X, \tag{56}$$

which is exactly the marginal law in Eq. (4). Therefore $\widetilde{q}_X(X_t \mid X_0) = q_X(X_t \mid X_0)$. □

Proposition 2 (Adjacency-branch same-marginal property). *Let $\epsilon^A \sim \mathcal{N}_{\text{sym}}(0, I)$ and define*

$$\widetilde{q}_A(A_{1:T} \mid A_0, \epsilon^A) = \prod_{s=1}^{T} \delta\Big(A_s - \sqrt{\bar{\alpha}_s^A} A_0 - \sqrt{1 - \bar{\alpha}_s^A} \, \epsilon^A\Big). \tag{57}$$

Then for every $t \in \{1, \dots, T\}$,

$$\widetilde{q}_A(A_t \mid A_0) := \int \widetilde{q}_A(A_{1:T} \mid A_0, \epsilon^A) \mathcal{N}_{\text{sym}}(\epsilon^A; 0, I) \, dA_{1:t-1} \, dA_{t+1:T} \, d\epsilon^A = q_A(A_t \mid A_0), \tag{58}$$

where $q_A(A_t \mid A_0)$ is the Gaussian marginal in Eq. (7).

Proof. Fix t. Integrating out A_s for $s \neq t$ yields

$$\widetilde{q}_A(A_t \mid A_0) = \int \delta\Big(A_t - \sqrt{\bar{\alpha}_t^A} A_0 - \sqrt{1 - \bar{\alpha}_t^A} \, \epsilon^A\Big) \mathcal{N}_{\text{sym}}(\epsilon^A; 0, I) \, d\epsilon^A. \tag{59}$$

Thus $\widetilde{q}_A(A_t \mid A_0)$ is the law of the affine transform $\sqrt{\bar{\alpha}_t^A} A_0 + \sqrt{1 - \bar{\alpha}_t^A} \, \epsilon^A$. Since $\epsilon^A \sim \mathcal{N}_{\text{sym}}(0, I)$, this law is exactly

$$\mathcal{N}_{\text{sym}}\Big(\sqrt{\bar{\alpha}_t^A} A_0, (1 - \bar{\alpha}_t^A) I\Big), \tag{60}$$

which is the marginal in Eq. (7). Therefore $\widetilde{q}_A(A_t \mid A_0) = q_A(A_t \mid A_0)$. □

Proposition 3 (Joint graph-state same-marginal property). *Let $q_t(G_t \mid G_0)$ denote the time-t conditional marginal induced by the forward process in Eq. (1). Assume:*

1. *the forward path factorizes across branches,*

$$q(G_{1:T} \mid G_0) = q_X(X_{1:T} \mid X_0) \, q_A(A_{1:T} \mid A_0), \tag{61}$$

equivalently, the feature and adjacency forward noises are independent;

2. *the auxiliary law of the same-marginal family factorizes across branches as*

$$q_\Lambda(\Lambda_{1:T})\,\mathcal{N}(z^X;0,I)\,\mathcal{N}_{\mathrm{sym}}(\epsilon^A;0,I), \tag{62}$$

with z^X independent of $\Lambda_{1:T}$ and of ϵ^A;

3. *for every t, the scale-path marginal satisfies $q_{\Lambda,t} = p_\Lambda$.*

Define the joint deterministic family by

$$\widetilde{q}(G_{1:T} \mid G_0, \Lambda_{1:T}, z^X, \epsilon^A) := \widetilde{q}_X(X_{1:T} \mid X_0, \Lambda_{1:T}, z^X)\,\widetilde{q}_A(A_{1:T} \mid A_0, \epsilon^A). \tag{63}$$

Then for every $t \in \{1,\dots,T\}$,

$$\begin{aligned}\widetilde{q}(G_t \mid G_0) := \int \widetilde{q}(G_{1:T} \mid G_0, \Lambda_{1:T}, z^X, \epsilon^A) q_\Lambda(\Lambda_{1:T})\,\mathcal{N}(z^X;0,I)\mathcal{N}_{\mathrm{sym}}(\epsilon^A;0,I) \\ dG_{1:t-1}\,dG_{t+1:T}\,d\Lambda_{1:T}\,dz^X\,d\epsilon^A = q_t(G_t \mid G_0).\end{aligned} \tag{64}$$

Consequently, the induced noisy graph marginal also agrees with the original forward process:

$$\widetilde{p}_t(X_t, A_t) := \int p_{\mathrm{data}}(G_0)\,\widetilde{q}(G_t \mid G_0)\,dG_0 = \int p_{\mathrm{data}}(G_0)\,q_t(G_t \mid G_0)\,dG_0 = p_t(X_t, A_t). \tag{65}$$

Proof. By the branchwise factorization of the forward path,

$$q(G_{1:T} \mid G_0) = \Big(\prod_{s=1}^{T} q_X(X_s \mid X_{s-1})\Big)\Big(\prod_{s=1}^{T} q_A(A_s \mid A_{s-1})\Big) = q_X(X_{1:T} \mid X_0)\,q_A(A_{1:T} \mid A_0). \tag{66}$$

Integrating out all variables except (X_t, A_t) therefore gives

$$q_t(G_t \mid G_0) = q_X(X_t \mid X_0)\,q_A(A_t \mid A_0). \tag{67}$$

On the deterministic side, assumption 2 implies that the auxiliary variables factorize across the feature and adjacency branches, so the induced one-time marginal of the deterministic family also factorizes:

$$\begin{aligned}\widetilde{q}(G_t \mid G_0) &= \widetilde{q}_X(X_t \mid X_0)\,\widetilde{q}_A(A_t \mid A_0) &(68)\\ &= q_X(X_t \mid X_0)\,q_A(A_t \mid A_0), &(69)\end{aligned}$$

where the second line uses Propositions 1 and 2. Hence $\widetilde{q}(G_t \mid G_0) = q_t(G_t \mid G_0)$ for every t. Integrating both sides against $p_{\mathrm{data}}(G_0)$ yields the equality of the induced noisy graph marginals, $\widetilde{p}_t(X_t, A_t) = p_t(X_t, A_t)$. □

C.1. Deterministic Reverse Updates from the Same-Marginal Family

This subsection derives the deterministic updates in Eqs. (22) to (24). The key point is that these updates are not additional modeling assumptions: they follow by algebraic elimination inside the same-marginal non-Markovian families introduced in the main text. The stochastic reverse factors $r^X_{\theta,t}$ and $r^A_{\phi,t}$ used in the variational objective are therefore conceptually distinct from the deterministic sampler derived below.

Feature branch: oracle derivation. Condition on a scale path $\Lambda_{1:T}$ and a shared Gaussian latent $z^X \sim \mathcal{N}(0, I)$. By definition of the same-marginal family, for any time index $u \in \{1, \dots, T\}$,

$$X_u = \bar{\gamma}_u^X X_0 + \bar{\sigma}_u^X \Lambda_u^{1/2} z^X. \tag{70}$$

Fix two times $s < t$. Evaluating Eq. (70) at t and solving for X_0 gives

$$X_0 = \frac{X_t - \bar{\sigma}_t^X \Lambda_t^{1/2} z^X}{\bar{\gamma}_t^X}. \tag{71}$$

Substituting Eq. (71) into the same family at time s yields

$$X_s = \bar{\gamma}_s^X \left(\frac{X_t - \bar{\sigma}_t^X \Lambda_t^{1/2} z^X}{\bar{\gamma}_t^X} \right) + \bar{\sigma}_s^X \Lambda_s^{1/2} z^X \tag{72}$$

$$= \frac{\bar{\gamma}_s^X}{\bar{\gamma}_t^X} X_t + \left(\bar{\sigma}_s^X \Lambda_s^{1/2} - \frac{\bar{\gamma}_s^X}{\bar{\gamma}_t^X} \bar{\sigma}_t^X \Lambda_t^{1/2} \right) z^X. \tag{73}$$

Equivalently, defining the oracle clean estimate

$$\widehat{X}_0^{\text{oracle}} := \frac{X_t - \bar{\sigma}_t^X \Lambda_t^{1/2} z^X}{\bar{\gamma}_t^X}, \tag{74}$$

we can write the deterministic reverse update in the compact form

$$X_s = \bar{\gamma}_s^X \widehat{X}_0^{\text{oracle}} + \bar{\sigma}_s^X \Lambda_s^{1/2} z^X. \tag{75}$$

Hence the deterministic map from X_t to X_s is obtained by eliminating X_0 between two members of the same-marginal family; it is not an ad hoc definition.

Importantly, the coefficient at time s is Λ_s, not Λ_t. This is forced by the pathwise family Eq. (70): the state at time s is parameterized by $(\bar{\gamma}_s^X, \bar{\sigma}_s^X, \Lambda_s)$. Replacing Λ_s by Λ_t would define a different non-Markovian family and therefore a different deterministic sampler. In particular, for the one-step update $s = t - 1$, the correct feature update uses Λ_{t-1}, exactly as stated in the main text.

Adjacency branch: oracle derivation. The Gaussian same-marginal family on adjacency is

$$A_u = \sqrt{\bar{\alpha}_u^A} A_0 + \sqrt{1 - \bar{\alpha}_u^A}\, \epsilon^A, \qquad \epsilon^A \sim \mathcal{N}_{\text{sym}}(0, I), \tag{76}$$

with a shared latent ϵ^A across time. Evaluating Eq. (76) at time t and solving for A_0 gives

$$A_0 = \frac{A_t - \sqrt{1 - \bar{\alpha}_t^A}\, \epsilon^A}{\sqrt{\bar{\alpha}_t^A}}. \tag{77}$$

Substituting Eq. (77) into the same family at time $s < t$ yields

$$A_s = \sqrt{\bar{\alpha}_s^A} \left(\frac{A_t - \sqrt{1 - \bar{\alpha}_t^A}\, \epsilon^A}{\sqrt{\bar{\alpha}_t^A}} \right) + \sqrt{1 - \bar{\alpha}_s^A}\, \epsilon^A \tag{78}$$

$$= \sqrt{\frac{\bar{\alpha}_s^A}{\bar{\alpha}_t^A}}\, A_t + \left(\sqrt{1 - \bar{\alpha}_s^A} - \sqrt{\frac{\bar{\alpha}_s^A}{\bar{\alpha}_t^A}} \sqrt{1 - \bar{\alpha}_t^A} \right) \epsilon^A. \tag{79}$$

Equivalently, with the oracle clean estimate

$$\widehat{A}_0^{\text{oracle}} := \frac{A_t - \sqrt{1 - \bar{\alpha}_t^A}\, \epsilon^A}{\sqrt{\bar{\alpha}_t^A}}, \tag{80}$$

we obtain the compact deterministic update

$$A_s = \sqrt{\bar{\alpha}_s^A}\, \widehat{A}_0^{\text{oracle}} + \sqrt{1 - \bar{\alpha}_s^A}\, \epsilon^A. \tag{81}$$

Again, this is simply the result of eliminating A_0 inside the Gaussian same-marginal family.

Replacing oracle latents by network predictions. At sampling time the oracle latents (z^X, ϵ^A) are unavailable. We therefore use the coupled denoisers

$$\hat{z}_t^X = \mathcal{D}_\theta^X(X_t, A_t, \Lambda_{1:t}, t), \qquad \hat{\epsilon}_t^A = \mathcal{D}_\phi^A(X_t, A_t, \Lambda_{1:t}, t) \tag{82}$$

as estimators of the shared latent variables of the same-marginal family. Replacing (z^X, ϵ^A) in Eqs. (74), (75), (80) and (81) by $(\hat{z}_t^X, \hat{\epsilon}_t^A)$ yields exactly the deterministic sampling formulas used in the main text:

$$\hat{X}_0 = \frac{X_t - \bar{\sigma}_t^X \Lambda_t^{1/2} \hat{z}_t^X}{\bar{\gamma}_t^X}, \qquad \hat{A}_0 = \frac{A_t - \sqrt{1-\bar{\alpha}_t^A}\,\hat{\epsilon}_t^A}{\sqrt{\bar{\alpha}_t^A}}, \tag{83}$$

$$X_s = \bar{\gamma}_s^X \hat{X}_0 + \bar{\sigma}_s^X \Lambda_s^{1/2} \hat{z}_t^X, \qquad A_s = \sqrt{\bar{\alpha}_s^A}\,\hat{A}_0 + \sqrt{1-\bar{\alpha}_s^A}\,\hat{\epsilon}_t^A. \tag{84}$$

Thus the only approximation in the deterministic sampler is the substitution of unknown oracle latents by network predictions; the algebraic form of the update is exact.

If one instead parameterized the feature denoiser to predict the scaled latent $\eta_t^X := \Lambda_t^{1/2} z^X$ rather than z^X itself, the equivalent feature update would be

$$X_s = \bar{\gamma}_s^X \hat{X}_0 + \bar{\sigma}_s^X \left(\frac{\Lambda_s}{\Lambda_t}\right)^{1/2} \hat{\eta}_t^X. \tag{85}$$

Our main-text parameterization predicts z^X directly, so the correct form in Eq. (23) is the one with $\Lambda_s^{1/2} \hat{z}_t^X$.

Remark: relation to implicit same-marginal samplers. The stochastic reverse model in Eqs. (14), (17) and (18) belongs to the exact augmented Markov process used for variational training. The deterministic updates derived above are not those stochastic reverse kernels with the noise term removed. Instead, they come from a separate non-Markovian same-marginal family obtained by fixing shared latent variables across time and eliminating the clean state by algebra. This is exactly the DDPM/DDIM separation on the Gaussian adjacency branch. The feature branch is the corresponding stable-family analogue: after conditioning on the scale path, one fixes a shared Gaussian latent z^X across time and obtains a deterministic implicit sampler of the same flavor. In other words, training-time stochastic reverse and sampling-time deterministic same-marginal transport are two different objects, even though they reuse the same denoisers.

D. From the Exact ELBO to the Deployed Denoising Proxy

The main text defines an exact negative ELBO for the *augmented Markov chain* (13)–(14). The implementation, however, is not the line-by-line optimization of that ELBO in its original Gaussian posterior coordinates. The reason is that the exact ELBO of the augmented feature chain is naturally written in the *pathwise Gaussian coordinates* induced by conditioning on the full scale history $\Lambda_{1:t}$, whereas the deployed sampler and denoiser are written in the *same-marginal coordinates* $\bar{\sigma}_t^X \Lambda_t^{1/2}$ used in (10), (22), and (23). This appendix separates these two layers explicitly:

1. we first rewrite the exact ELBO as a weighted denoising loss in the exact pathwise Gaussian coordinates;

2. we then pass to the same-marginal feature coordinates used by the deployed implementation via an explicit scaling identity; and

3. we finally define the practical proxy loss as a reweighted version of that exact surrogate.

A note on the feature marginal. For the exact augmented Markov chain, $q_X(X_t \mid X_0, \Lambda_{1:t})$ depends on the *full* scale history $\Lambda_{1:t}$ through a pathwise variance, not only on Λ_t. The one-level representation $X_t = \bar{\gamma}_t^X X_0 + \bar{\sigma}_t^X \Lambda_t^{1/2} z_t^X$ belongs to the same-marginal family used later by the deterministic sampler. The distinction is immaterial for sampling, but it is essential for an exact ELBO proof.

Feature pathwise variance. For $t \geq 1$, define the conditional variance of the Gaussianized feature chain

$$\left(v_t^X(\Lambda_{1:t})\right)^2 := \sum_{i=1}^{t} \left(\sigma_i^X \prod_{j=i+1}^{t} \gamma_j^X \right)^2 \Lambda_i, \qquad v_0^X := 0. \tag{86}$$

Unrolling the linear conditional-Gaussian chain gives

$$X_t = \bar{\gamma}_t^X X_0 + v_t^X(\Lambda_{1:t})\, \eta_t^X, \qquad \eta_t^X \sim \mathcal{N}(0, I), \tag{87}$$

conditioned on $(X_0, \Lambda_{1:t})$. For the adjacency branch, we retain the standard DDPM marginal

$$A_t = \sqrt{\bar{\alpha}_t^A} A_0 + \sqrt{1 - \bar{\alpha}_t^A}\, \epsilon^A, \qquad \epsilon^A \sim \mathcal{N}_{\text{sym}}(0, I). \tag{88}$$

D.1. Exact ELBO-Denoising Identity

We first characterize the exact posteriors appearing in (17) and (18).

Feature posterior. Conditioned on $(X_t, X_0, \Lambda_{1:t})$, the feature posterior is Gaussian:

$$q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t}) = \mathcal{N}\big(\widetilde{\mu}_t^X, (\widetilde{\nu}_t^X)^2 I\big)\,, \tag{89}$$

with

$$(\widetilde{\nu}_t^X)^2 = \frac{(\sigma_t^X)^2 \Lambda_t \, (v_{t-1}^X(\Lambda_{1:t-1}))^2}{(v_t^X(\Lambda_{1:t}))^2}, \tag{90}$$

and posterior mean

$$\widetilde{\mu}_t^X = \frac{(\sigma_t^X)^2 \Lambda_t}{(v_t^X)^2}\, \bar{\gamma}_{t-1}^X X_0 + \frac{\gamma_t^X (v_{t-1}^X)^2}{(v_t^X)^2}\, X_t, \tag{91}$$

where, to lighten notation, v_t^X and v_{t-1}^X denote $v_t^X(\Lambda_{1:t})$ and $v_{t-1}^X(\Lambda_{1:t-1})$, respectively. Using (87), this can be rewritten in noise coordinates as

$$\widetilde{\mu}_t^X = \frac{1}{\gamma_t^X} \left(X_t - \frac{(\sigma_t^X)^2 \Lambda_t}{v_t^X}\, \eta_t^X \right). \tag{92}$$

Adjacency posterior. Conditioned on (A_t, A_0), the adjacency posterior is the usual DDPM posterior

$$q_A(A_{t-1} \mid A_t, A_0) = \mathcal{N}_{\text{sym}}\Big(\widetilde{\mu}_t^A, \widetilde{\beta}_t^A I\Big)\,, \tag{93}$$

with

$$\widetilde{\beta}_t^A = \frac{\beta_t^A (1 - \bar{\alpha}_{t-1}^A)}{1 - \bar{\alpha}_t^A}, \tag{94}$$

and

$$\widetilde{\mu}_t^A = \frac{1}{\sqrt{\alpha_t^A}} \left(A_t - \frac{\beta_t^A}{\sqrt{1 - \bar{\alpha}_t^A}}\, \epsilon^A \right). \tag{95}$$

We now parameterize the training-time reverse families by denoising heads acting on the common conditioning variable

$$U_t := (X_t, A_t, \Lambda_{1:t}, t). \tag{96}$$

Let the feature head $g^X_{\theta,t}(U_t)$ and the adjacency head $g^A_{\phi,t}(U_t)$ define reverse means through

$$r^X_{\theta,t}(X_{t-1} \mid U_t) = \mathcal{N}\big(\mu^X_{\theta,t}(U_t), \rho^X_t(\Lambda_{1:t})I\big), \qquad \mu^X_{\theta,t}(U_t) := \frac{1}{\gamma^X_t}\left(X_t - \frac{(\sigma^X_t)^2\Lambda_t}{v^X_t}\, g^X_{\theta,t}(U_t)\right), \tag{97}$$

and

$$r^A_{\phi,t}(A_{t-1} \mid U_t) = \mathcal{N}_{\mathrm{sym}}\big(\mu^A_{\phi,t}(U_t), \rho^A_t I\big), \qquad \mu^A_{\phi,t}(U_t) := \frac{1}{\sqrt{\alpha^A_t}}\left(A_t - \frac{\beta^A_t}{\sqrt{1-\bar\alpha^A_t}}\, g^A_{\phi,t}(U_t)\right), \tag{98}$$

where $\rho^X_t(\Lambda_{1:t}) > 0$ and $\rho^A_t > 0$ are fixed reverse variances.

Define the exact weighted denoising surrogate

$$\mathcal{L}^{\mathrm{exact}}_{\mathrm{den}} := \mathbb{E}_{t\sim\pi}\left[\frac{\mathbf{1}_{\{t\geq 2\}}}{\pi(t)}\mathbb{E}_q\Big[\omega^X_t(\Lambda_{1:t})\,\|g^X_{\theta,t}(U_t) - \eta^X_t\|_F^2 + \omega^A_t\,\|g^A_{\phi,t}(U_t) - \epsilon^A\|_F^2\Big]\right], \tag{99}$$

with weights

$$\omega^X_t(\Lambda_{1:t}) := \frac{\big((\sigma^X_t)^2\Lambda_t\big)^2}{2\rho^X_t(\Lambda_{1:t})(\gamma^X_t)^2(v^X_t(\Lambda_{1:t}))^2}, \qquad \omega^A_t := \frac{(\beta^A_t)^2}{2\rho^A_t\alpha^A_t(1-\bar\alpha^A_t)}. \tag{100}$$

Proposition 4 (Exact ELBO-denoising identity). *Assume that the reverse variances in (97) and (98) are fixed and do not depend on the network outputs. Then the variational objective in (17)–(18) satisfies*

$$\mathcal{L}^{\mathrm{VI}}_X + \mathcal{L}^{\mathrm{VI}}_A = \mathcal{L}^{\mathrm{exact}}_{\mathrm{den}} + C_{\mathrm{ELBO}}, \tag{101}$$

where C_{ELBO} is independent of (θ, ϕ).

Proof. We use the Gaussian KL identity

$$D_{\mathrm{KL}}\big(\mathcal{N}(m_1, s_1^2 I)\,\|\,\mathcal{N}(m_2, s_2^2 I)\big) = \frac{\|m_1 - m_2\|_F^2}{2s_2^2} + C(s_1^2, s_2^2), \tag{102}$$

where the second term is independent of m_2.

For the feature branch, combining (92) and (97) gives

$$\widetilde\mu^X_t - \mu^X_{\theta,t}(U_t) = \frac{(\sigma^X_t)^2\Lambda_t}{\gamma^X_t v^X_t}\big(g^X_{\theta,t}(U_t) - \eta^X_t\big). \tag{103}$$

Therefore,

$$D_{\mathrm{KL}}\big(q_X(X_{t-1} \mid X_t, X_0, \Lambda_{1:t})\,\|\,r^X_{\theta,t}(X_{t-1} \mid U_t)\big) = \omega^X_t(\Lambda_{1:t})\,\|g^X_{\theta,t}(U_t) - \eta^X_t\|_F^2 + C^X_t, \tag{104}$$

with C^X_t independent of θ.

Likewise, using (95) and (98),

$$\widetilde\mu^A_t - \mu^A_{\phi,t}(U_t) = \frac{\beta^A_t}{\sqrt{\alpha^A_t(1-\bar\alpha^A_t)}}\big(g^A_{\phi,t}(U_t) - \epsilon^A\big), \tag{105}$$

and hence

$$D_{\mathrm{KL}}\big(q_A(A_{t-1} \mid A_t, A_0)\,\|\,r^A_{\phi,t}(A_{t-1} \mid U_t)\big) = \omega^A_t\,\|g^A_{\phi,t}(U_t) - \epsilon^A\|_F^2 + C^A_t, \tag{106}$$

with C^A_t independent of ϕ. Substituting (104) and (106) into (17) and (18), and collecting all terms independent of (θ, ϕ) into C_{ELBO}, proves (101). □

Matched-variance special case. If the reverse variances are set to the exact posterior variances, $\rho_t^X(\Lambda_{1:t}) = (\widetilde{\nu}_t^X)^2$ and $\rho_t^A = \widetilde{\beta}_t^A$, then the weights simplify to

$$\omega_t^X(\Lambda_{1:t}) = \frac{(\sigma_t^X)^2\Lambda_t}{2(\gamma_t^X)^2(v_{t-1}^X(\Lambda_{1:t-1}))^2}, \qquad \omega_t^A = \frac{\beta_t^A}{2\alpha_t^A(1-\bar{\alpha}_{t-1}^A)}. \tag{107}$$

This is the exact DDPM-style "KL equals weighted denoising mean squared error (MSE)" reduction for the augmented heterogeneous process.

D.2. From the Exact Denoising Surrogate to the Deployed Proxy

The implementation is written in the same-marginal feature coordinates used by the deterministic sampler. We therefore introduce the practical feature target

$$\zeta_t^X := \frac{X_t - \bar{\gamma}_t^X X_0}{\bar{\sigma}_t^X \Lambda_t^{1/2}}, \tag{108}$$

which satisfies the exact scaling relation

$$\zeta_t^X = \underbrace{\frac{v_t^X(\Lambda_{1:t})}{\bar{\sigma}_t^X \Lambda_t^{1/2}}}_{=:\, c_t^X(\Lambda_{1:t})} \eta_t^X. \tag{109}$$

Note that under the exact augmented Markov chain, ζ_t^X is generally *not* standard Gaussian; it is the natural same-marginal target associated with the deployed feature parameterization.

For any exact latent predictor $g_{\theta,t}^X(U_t)$, define the corresponding practical predictor

$$h_{\theta,t}^X(U_t) := c_t^X(\Lambda_{1:t})\, g_{\theta,t}^X(U_t). \tag{110}$$

Proposition 5 (Exact coordinate-change identity for the feature proxy). *For every predictor pair $(g_{\theta,t}^X, h_{\theta,t}^X)$ linked by (110),*

$$\omega_t^X(\Lambda_{1:t})\|g_{\theta,t}^X(U_t) - \eta_t^X\|_F^2 = \omega_t^X(\Lambda_{1:t})\,(c_t^X(\Lambda_{1:t}))^{-2}\|h_{\theta,t}^X(U_t) - \zeta_t^X\|_F^2. \tag{111}$$

Proof. By (109) and (110),

$$h_{\theta,t}^X(U_t) - \zeta_t^X = c_t^X(\Lambda_{1:t})\big(g_{\theta,t}^X(U_t) - \eta_t^X\big). \tag{112}$$

Taking squared Frobenius norms and multiplying by $\omega_t^X(\Lambda_{1:t})(c_t^X(\Lambda_{1:t}))^{-2}$ yields (111). □

Proposition 5 is the precise sense in which the feature-side denoising objective used by the deployed implementation is a coordinate change of the exact ELBO-derived surrogate: the target is rewritten in the same-marginal coordinates of the sampler, and the exact ELBO weight transforms accordingly.

We may therefore write an exact same-marginal surrogate for the two branches as

$$\mathcal{L}_{\text{den}}^{\text{sm}} := \mathbb{E}_{t\sim\pi}\left[\frac{\mathbf{1}_{\{t\geq 2\}}}{\pi(t)}\mathbb{E}_q\left[\widetilde{\omega}_t^X(\Lambda_{1:t})\|h_{\theta,t}^X(U_t) - \zeta_t^X\|_F^2 + \omega_t^A\|g_{\phi,t}^A(U_t) - \epsilon^A\|_F^2\right]\right], \tag{113}$$

where

$$\widetilde{\omega}_t^X(\Lambda_{1:t}) := \omega_t^X(\Lambda_{1:t})\,(c_t^X(\Lambda_{1:t}))^{-2}. \tag{114}$$

By Proposition 5, $\mathcal{L}_{\text{den}}^{\text{sm}}$ is still exactly equivalent to the ELBO up to the additive constant C_{ELBO}.

Deployed proxy. In practice we typically replace the exact ELBO weights $\widetilde{\omega}_t^X(\Lambda_{1:t})$ and ω_t^A by simpler positive weights λ_t^X and λ_t^A, obtaining

$$\mathcal{L}_{\text{proxy}} := \mathbb{E}_{t\sim\pi}\left[\frac{\mathbf{1}_{\{t\geq 2\}}}{\pi(t)}\mathbb{E}_q\Big[\lambda_t^X(\Lambda_{1:t})\|h_{\theta,t}^X(U_t)-\zeta_t^X\|_F^2+\lambda_t^A\|g_{\phi,t}^A(U_t)-\epsilon^A\|_F^2\Big]\right]. \tag{115}$$

When

$$\lambda_t^X(\Lambda_{1:t})=\widetilde{\omega}_t^X(\Lambda_{1:t}),\qquad \lambda_t^A=\omega_t^A, \tag{116}$$

we recover the exact ELBO-derived denoising surrogate. For other positive choices, (115) is a reweighted proxy of the exact ELBO surrogate.

Proposition 6 (Same Conditional Regression Targets Under Positive Reweighting). *Assume that $\lambda_t^X(\Lambda_{1:t})>0$ and $\lambda_t^A>0$ are measurable with respect to the conditioning variable U_t and do not depend on the network outputs. Then the Bayes-optimal predictors of (115) satisfy*

$$h_{\theta,t}^{X,\star}(U_t)=\mathbb{E}[\zeta_t^X\mid U_t],\qquad g_{\phi,t}^{A,\star}(U_t)=\mathbb{E}[\epsilon^A\mid U_t]. \tag{117}$$

In particular, because $\zeta_t^X=c_t^X\eta_t^X$ with $c_t^X(\Lambda_{1:t})$ measurable in U_t, the feature predictor in (115) targets the same posterior mean as the exact ELBO surrogate, expressed in the implementation coordinates.

Proof. Condition on $U_t=u$. Since $\lambda_t^X(u)$ and $\lambda_t^A(u)$ are positive constants with respect to the optimization over the predictor values at u, minimizing

$$\lambda_t^X(u)\,\mathbb{E}\big[\|h-\zeta_t^X\|_F^2\mid U_t=u\big] \tag{118}$$

with respect to h is equivalent to minimizing the unweighted conditional squared error, whose unique minimizer is $\mathbb{E}[\zeta_t^X\mid U_t=u]$. The same argument gives the adjacency predictor $\mathbb{E}[\epsilon^A\mid U_t=u]$. □

Therefore, the practical loss used in experiments should be interpreted as follows:

1. the exact ELBO of the augmented heterogeneous diffusion is first rewritten as a weighted denoising objective in the exact pathwise Gaussian coordinates;
2. the feature target is then moved to the same-marginal coordinates of the deployed sampler by an explicit scale change; and
3. the final training objective is a positive reweighting of that exact same-marginal surrogate, preserving the denoising targets while simplifying implementation.

This is the precise sense in which the deployed denoising loss is a *proxy* for the ELBO: it is exact up to a coordinate change and, when the exact weights are retained, identical to the ELBO-derived denoising surrogate up to constants; when simpler weights are used, it becomes a reweighted surrogate with the same conditional regression targets.

E. Controlled Benchmark Definition

This appendix gives a reproducible definition of the controlled synthetic benchmark used in Section 4.1. We separate the benchmark into two protocols: joint generation from noise, and conditional adjacency generation under a fixed role assignment. Throughout, $X=(X_1,\dots,X_N)$ denotes a role assignment over $N\in\{12,\dots,16\}$ active nodes with

$$X_i\in\mathcal{R}:=\{\texttt{common},\texttt{bridge},\texttt{rare\_hub}\},$$

and $A\in\{0,1\}^{N\times N}$ is a symmetric binary adjacency matrix with zero diagonal.

E.1. Shared Toy Oracle and Validity Set

A graph (X, A) is valid if and only if all of the following conditions hold.

1. Every active node has degree in $[1, 5]$.
2. The induced subgraph on the `common` nodes decomposes into connected components of size $2, 3$, or 4, and each such component is either a chain or a clique.
3. Every `bridge` node connects only to `common` nodes, has degree $2, 3$, or 4, and its neighbors lie in at least two distinct common-node components.
4. Every `rare_hub` node connects only to `common` nodes, has degree 2 or 3, and no two of its neighbors are adjacent.

For a fixed role assignment X, the valid adjacency set is therefore

$$\mathcal{V}(X) := \{A \in \{0,1\}^{N\times N} : A = A^\top,\ \mathrm{diag}(A) = 0,\ \mathrm{Valid}(X, A) = 1\}. \tag{119}$$

After constructing a valid graph, the toy oracle may apply the legality-preserving one-step rewiring defined in Appendix E.4; this makes the conditional law $A \mid X$ multimodal without changing the validity rules above.

E.2. Protocol A: Joint Generation Metrics

For joint generation, the reference set $\mathcal{D}_{\mathrm{ref}}$ is the toy test split and the generated set $\mathcal{D}_{\mathrm{gen}}$ is the model sample set produced from noise under the same decoding pipeline used for the reported results.

Role-pair edge vector. Let

$$\mathcal{P} := \left\{ \begin{matrix} (\texttt{common}, \texttt{common}), (\texttt{common}, \texttt{bridge}), (\texttt{common}, \texttt{rare\_hub}), \\ (\texttt{bridge}, \texttt{bridge}), (\texttt{bridge}, \texttt{rare\_hub}), (\texttt{rare\_hub}, \texttt{rare\_hub}) \end{matrix} \right\}$$

be the unordered role-pair index set. For each graph (X, A), define the role-pair edge count vector $\psi(X, A) \in \mathbb{R}^6$ by

$$\psi_{rs}(X, A) := \sum_{1\le i<j\le N} A_{ij}\, \mathbf{1}[\{X_i, X_j\} = \{r, s\}], \qquad (r, s) \in \mathcal{P}. \tag{120}$$

Joint statistic vector and joint error. Define the role-count vector

$$c(X) := \big(n_{\texttt{common}}(X), n_{\texttt{bridge}}(X), n_{\texttt{rare\_hub}}(X)\big) \in \mathbb{R}^3.$$

Let $h(X, A) \in \mathbb{R}^3$ be the histogram of common-only connected components of sizes $2, 3, 4$, let $m(X, A) \in \mathbb{R}^4$ collect

1. the number of common-node chain components,
2. the number of common-node clique components,
3. the mean number of distinct common-node components spanned by a `bridge` node, and
4. the mean degree of `rare_hub` nodes,

and let

$$e(A) := \sum_{1 \le i < j \le N} A_{ij}$$

be the total undirected edge count. The joint statistic vector is

$$\phi(X, A) := \big(c(X), \psi(X, A), h(X, A), m(X, A), e(A)\big) \in \mathbb{R}^{17}. \tag{121}$$

The reported joint error is the mean absolute deviation between the empirical means of these statistics:

$$\mathrm{JointError}(\mathcal{D}_{\mathrm{ref}}, \mathcal{D}_{\mathrm{gen}}) := \frac{1}{17} \left\| \frac{1}{|\mathcal{D}_{\mathrm{ref}}|} \sum_{(X,A) \in \mathcal{D}_{\mathrm{ref}}} \phi(X, A) - \frac{1}{|\mathcal{D}_{\mathrm{gen}}|} \sum_{(X,A) \in \mathcal{D}_{\mathrm{gen}}} \phi(X, A) \right\|_1. \tag{122}$$

Role-pair maximum mean discrepancy (MMD). Given role-pair vectors $\{\psi_i\}_{i=1}^n$ and $\{\widehat{\psi}_j\}_{j=1}^m$, we use the Gaussian-kernel MMD with bandwidth $\sigma = 1$:

$$\mathrm{MMD}_k^2(\{\psi_i\}, \{\widehat{\psi}_j\}) = \frac{1}{n^2} \sum_{i,i'=1}^{n} k(\psi_i, \psi_{i'}) + \frac{1}{m^2} \sum_{j,j'=1}^{m} k(\widehat{\psi}_j, \widehat{\psi}_{j'}) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(\psi_i, \widehat{\psi}_j), \tag{123}$$

$$k(u, v) := \exp\left(-\frac{\|u - v\|_2^2}{2\sigma^2} \right), \qquad \sigma = 1.$$

The Protocol A role-pair MMD is obtained by applying (123) to $\{\psi(X, A) : (X, A) \in \mathcal{D}_{\mathrm{ref}}\}$ and $\{\psi(X, A) : (X, A) \in \mathcal{D}_{\mathrm{gen}}\}$. The same kernel and bandwidth are reused below for the fixed-X raw and repaired conditional role-pair MMDs. We report this squared discrepancy and refer to it as "role-pair MMD" for brevity.

E.3. Protocol B: Fixed-X Conditional Benchmark

The fixed-X benchmark clamps the role assignment and evaluates only the adjacency branch. Let $\mathcal{C}$ denote the selected set of held-out fixed role assignments. For each $X \in \mathcal{C}$, we draw

- an oracle set $\mathcal{A}_{\mathrm{oracle}}(X)$ of valid adjacencies sampled from the toy oracle under that X, and
- a model set $\mathcal{A}_\theta(X)$ of predicted adjacencies from the diffusion model under the same X.

Every fixed-X metric is computed casewise and then averaged over $\mathcal{C}$.

E.3.1. Fixed-X Case Selection

Candidate cases are drawn from the toy test split subject to two constraints:

1. the exact held-out role assignment X does not appear in the train split, and
2. a short oracle probe produces at least two distinct valid adjacencies under that same X, so that $A \mid X$ is non-degenerate for the case.

For each eligible X, define the role-count vector $c(X)$ as above and the common-node attachment pressure

$$\mathrm{pressure}(X) := \frac{2(n_{\texttt{bridge}}(X) + n_{\texttt{rare\_hub}}(X))}{5\, n_{\texttt{common}}(X)}, \tag{124}$$

which measures how much common-node degree budget is consumed by bridge and rare-hub attachments under the shared degree cap 5. We also define the distance from the empirical held-out role-frequency mean

$$\text{rfreq\_dist}(X) := \left\| \frac{c(X)}{N} - \bar{p}_{\text{heldout}} \right\|_1, \tag{125}$$

where $\bar{p}_{\text{heldout}} \in \mathbb{R}^3$ is the mean normalized role-count vector over held-out cases.

The fixed-X selector then proceeds as follows.

1. Enumerate unique held-out role assignments from the test split and keep only the eligible cases above.
2. Compute pressure(X) and rfreq_dist(X) for each remaining case.
3. Form a `typical` pool consisting of cases with pressure in the interquartile band $[Q_{25}, Q_{75}]$, and a `boundary_hard` pool consisting of cases with pressure in $[Q_{75}, \infty)$.
4. Rank the `typical` cases by
$$|\text{pressure}(X) - \text{median\_pressure}| + \text{rfreq\_dist}(X) \tag{126}$$
in ascending order.
5. Rank the `boundary_hard` cases by
$$\text{pressure}(X) + 0.5\,\text{rfreq\_dist}(X) \tag{127}$$
in descending order.
6. Select a small diverse subset across role-count signatures $(N, c(X))$.

The compact screen used for Table 1 selects 2 `typical` + 2 `boundary_hard` cases, uses 8 oracle probe samples during case selection, and then evaluates each selected case with 64 oracle samples and 64 model samples.

E.3.2. Nearest-Valid Distance and Fixed-Role Repair

For a predicted adjacency A under a fixed role assignment X, the exact nearest-valid distance is

$$d_{\text{valid}}(A; X) := \min_{B \in \mathcal{V}(X)} \frac{1}{2}\|A - B\|_1. \tag{128}$$

Because A and B are symmetric binary adjacency matrices with zero diagonal, $\frac{1}{2}\|A-B\|_1$ is exactly the number of undirected edge flips needed to transform A into B.

The reported benchmark uses a candidate-search approximation to (128). Let $\mathcal{B}(X)$ be the candidate pool

$$\mathcal{B}(X) := \{A^{(1)}_{\text{oracle}}(X), \ldots, A^{(M)}_{\text{oracle}}(X)\} \tag{129}$$

augmented by a canonical valid graph $A_{\text{canon}}(X)$ whenever such a canonical construction exists. We then define $\widehat{\Pi}_{\text{valid}}(A \mid X)$ to be the candidate in $\mathcal{B}(X)$ that minimizes $\frac{1}{2}\|A - B\|_1$, where $A^{(1)}_{\text{oracle}}(X), \ldots, A^{(M)}_{\text{oracle}}(X)$ are valid oracle samples under the same fixed X. In the reported compact screen we use $M = 64$. The reported nearest-valid distance is therefore

$$\widehat{d}_{\text{valid}}(A; X) := \frac{1}{2}\left\| A - \widehat{\Pi}_{\text{valid}}(A \mid X) \right\|_1, \tag{130}$$

and the table reports the case-averaged mean

$$\text{NearestValidDistance} := \frac{1}{|\mathcal{C}|}\sum_{X\in\mathcal{C}}\frac{1}{|\mathcal{A}_\theta(X)|}\sum_{A\in\mathcal{A}_\theta(X)}\widehat{d}_{\text{valid}}(A;X). \tag{131}$$

No role projection is allowed in this benchmark: X is held fixed throughout repair. The repaired sample set is

$$\widetilde{\mathcal{A}}_\theta(X) := \{\widehat{\Pi}_{\text{valid}}(A \mid X) : A \in \mathcal{A}_\theta(X)\}. \tag{132}$$

E.3.3. Repaired Role-Pair Error and Repaired Role-Pair MMD

After fixed-role repair, the repaired role-pair error for a case X is the ℓ_1 deviation between the oracle and repaired mean role-pair vectors:

$$\text{RepairRolePairError}(X) := \frac{1}{6}\left\|\frac{1}{|\mathcal{A}_{\text{oracle}}(X)|}\sum_{A\in\mathcal{A}_{\text{oracle}}(X)}\psi(X,A) - \frac{1}{|\widetilde{\mathcal{A}}_\theta(X)|}\sum_{\widetilde{A}\in\widetilde{\mathcal{A}}_\theta(X)}\psi(X,\widetilde{A})\right\|_1. \tag{133}$$

The reported repaired role-pair error averages (133) over $X \in \mathcal{C}$.

The repaired role-pair MMD for a case X applies the same Gaussian-kernel MMD from (123) to the repaired and oracle role-pair vectors:

$$\text{RepairRolePairMMD}(X) := \text{MMD}_k^2\Big(\{\psi(X,A) : A \in \mathcal{A}_{\text{oracle}}(X)\}, \{\psi(X,\widetilde{A}) : \widetilde{A} \in \widetilde{\mathcal{A}}_\theta(X)\}\Big). \tag{134}$$

The table reports the average of $\text{RepairRolePairMMD}(X)$ over $X \in \mathcal{C}$.

E.4. Legality-Preserving One-Step Local Rewiring

The toy oracle introduces conditional multimodality by an optional one-step local rewiring of a valid graph. For a valid graph (X, A), call a pair of `common` nodes $\{u, v\}$ *protected* if there exists a `rare_hub` node h such that $A_{hu} = A_{hv} = 1$. Protected pairs may not be connected by a common-common edge.

The rewiring map is defined as follows.

1. Identify the set of common-node connected components whose induced subgraph is a chain and whose size is at least 3.
2. If no such component exists, return the current graph unchanged.
3. Otherwise, choose one such chain component C.
4. Let $\Pi_{\text{adm}}(C)$ be the set of permutations $\pi = (\pi_1, \ldots, \pi_{|C|})$ of the nodes in C for which the induced chain edge set

   $$E_\pi := \big\{\{\pi_1, \pi_2\}, \{\pi_2, \pi_3\}, \ldots, \{\pi_{|C|-1}, \pi_{|C|}\}\big\}$$

   differs from the current chain edge set on C and contains no protected pair. If $\Pi_{\text{adm}}(C) = \varnothing$, keep the current graph unchanged. Otherwise sample $\pi \in \Pi_{\text{adm}}(C)$.
5. Form a candidate adjacency A' by replacing the old common-common edges inside C by E_π and leaving all other edges unchanged.
6. Accept the rewiring if $\text{Valid}(X, A') = 1$; otherwise keep the original adjacency A.

This operation preserves the local chain motif class while changing the exact adjacency realization, which is the source of the conditional multiplicity in the fixed-X benchmark.

Table 6. Implementation details for the real-data experiments.

Backbone	GDSS-style coupled node/edge score networks with graph convolutional network (GCN) layers. Molecules: depth 2, hidden/attention 16, 3 layers, 3 linears, $(2, 8, 4)$, 4 heads. Generic graphs: depth 3, hidden/attention 32, 5 layers, 2 linears, $(2, 8, 4)$, 4 heads.
Max graph size	`max_node_num` $= 9$ for molecules and 20 for generic graphs.
Data representation	Molecules: atom one-hots and discrete bond-order adjacency. Generic graphs: degree one-hot node features and binary adjacency.
Molecule decoding	Quantize adjacency by $< 0.5 \mapsto 0$, $[0.5, 1.5) \mapsto 1$, $[1.5, 2.5) \mapsto 2$, and $\geq 2.5 \mapsto 3$. Binarize node features at 0.5, then decode and sanitize with the RDKit cheminformatics toolkit.
Generic-graph decoding	Threshold adjacency at 0.5. Remove self-loops and isolated nodes; if the graph becomes empty, insert one node.
Forward schedule	Molecules use variance-exploding (VE) schedules on both branches; generic graphs use variance-preserving (VP) schedules on both branches. Shared values: $\beta_{\min} = 0.1$, $\beta_{\max} = 1.0$, `num_scales` $= 1000$.
Heterogeneous setting	Feature branch: symmetric α-stable noise. Adjacency branch: Gaussian, ϵ-prediction, $\alpha_{\text{adj}} = 2.0$, and no extra beta correction terms. At $\alpha_x = 1.7$, both anchors use the Denoising Lévy Probabilistic Models (DLPM) Gaussian scale-mixture decomposition, per-entry stable scaling, and three-cap clipping 5/5/50. The retrained GDSS baseline uses Gaussian feature noise.
Training	Molecules: 300 epochs, batch size 1024. Generic graphs: 5000 epochs, batch size 128.
Optimization	Adam, learning-rate (LR) decay 0.999, exponential moving average (EMA) 0.999, weight decay 10^{-4}, gradient clip 1.0. Learning rate: 0.005 for molecules, 0.01 for generic graphs.
Sampling	Deterministic same-marginal reverse with $s = t - 1$, using Eqs. (23) and (24); no GDSS predictor-corrector or Langevin correction.
Evaluation	Molecules: validity without correction, Unique@10000, novelty, Fréchet ChemNet Distance (FCD), neighborhood subgraph pairwise distance kernel maximum mean discrepancy (NSPDK MMD). Generic graphs: degree, clustering, orbit, spectral, mean MMD.

F. Reproducibility Details for the Real-Data Experiments

This appendix summarizes one representative molecular setup and one generic-graph setup shared by community_small and ego_small. Both use the GDSS score-network backbone and masking conventions (Jo et al., 2022). For the main real-data tables, only the feature-noise family is varied, while the adjacency branch remains Gaussian; the bounded QM9 stable-adjacency check is reported separately below.

This appendix records the sampler configurations actually used in the reproduction runs. These experiments use the proposed forward–reverse pairing of the main method: denoising diffusion probabilistic model (DDPM)-style forward diffusion together with the same-marginal deterministic reverse, namely the stable feature update in Eq. (23) and the denoising diffusion implicit model (DDIM) update on the adjacency branch.

Implementation terminology. In Table 6, "per-entry stable scaling" means that the stable scale is sampled for each feature entry rather than shared isotropically across the whole feature vector, and "clamp 5/5/50" denotes a fixed three-cap clipping scheme in the stable decomposition that limits rare extreme draws during training and sampling.

Tail-index selection. The heterogeneous real-data experiments are feature-side ablations: α_{adj} is fixed at 2.0, α_x is anchored at 1.7, and the main tables report $\alpha_x \in \{2.0, 1.7, 1.5\}$. Molecules are selected mainly by FCD and NSPDK MMD, with validity without correction, Unique@10000, and novelty as secondary diagnostics. Generic graphs are selected by degree, clustering, orbit, spectral, and mean MMD. Additional molecule screens compare independent per-entry stable scaling versus isotropic shared scaling and 5/5/50 versus 10/10/100; generic graphs also test $\alpha_x \in \{1.5, 1.7, 1.9\}$, unclamped versus clamped stable scales, larger clamp thresholds, and independent per-entry versus isotropic shared scaling.

QM9 stable-adjacency check. The main real-data tables hold the adjacency branch Gaussian in order to test feature-side tail indices under a fixed structural noise family. We also ran a compute-limited QM9 check of stable-adjacency variants. This check is not a full real-data adjacency-family sweep: GS and SS were run on QM9, but complete GS/SS sweeps were not run on ZINC250k, community_small, or ego_small due to the available compute budget. We therefore use the QM9 runs only as a negative ablation against the main design choice, not as evidence that Gaussian adjacency is universally better on real data.

Table 7. QM9-only stable-adjacency check. The SG row is the reported Stable-X/Gaussian-A setting from Table 2. The GS and SS rows are additional stable-adjacency checks at 10,000 generated molecules; they are not part of a full multi-dataset GS/SS sweep. Best values within this check are in bold.

Variant	Validity w/o correction ↑	Unique@10000 ↑	Novelty ↑	FCD ↓	NSPDK MMD ↓
SG: Stable-X/Gaussian-A	**0.9726**	**0.9899**	0.862251	**2.778251**	**0.004516**
GS: Gaussian-X/Stable-A	0.9096	0.7257	0.794268	7.536593	0.037579
SS: Stable-X/Stable-A	0.9014	0.9036	**0.884462**	4.240119	0.014388

The GS run substantially degrades diversity and distributional metrics on QM9. The SS run improves over GS on several metrics, but it still does not improve over the reported SG setting except on novelty. This QM9 check is consistent with the controlled-probe interpretation: stable adjacency did not improve over the reported SG setting except on novelty. Since the check is limited to QM9, broader real-data stable-adjacency sweeps remain future work.

Compute budget. The real-data study is a bounded ablation, not a broad hyperparameter search. The Gaussian reference is a retrained GDSS checkpoint under the same pipeline. The molecular program is a single-seed branch sweep with 7 stable runs, main runs of 300 epochs, one 100-epoch provenance screen, and evaluation at 10,000 generated molecules. The generic-graph run ledger comes from the shared community_small/ego_small pipeline via the community_small ablation: 5 seed-42 screen variants, 10 clamp/mode grid points with 2 reused, and a 3-seed confirmation over 3 retained candidates, for 19 stable runs in total.