# MANIFOLD ALIGNMENT FOR GRAPH DIFFUSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Graph diffusion models are a powerful framework for generating complex structured data. However, existing training objectives such as denoising score matching or squared error minimization operate in a sample-wise manner, which optimize individual samples independently and overlook the underlying manifold geometry of the data distribution. To address this limitation, we propose manifold alignment learning (MAL) for graph diffusion models. MAL aligns the real and generated data distributions by non-parametric kernel density estimation on batch samples, while using a semantically informed embedding space solely to compute kernel similarities. By the MAL, we derive a consistency objective that encourages generated graphs to align with the real data manifold. Consequently, MAL promotes manifold-level distributional alignment at each diffusion step. Extensive experiments on graph benchmarks demonstrate that augmenting baseline graph diffusion models with MAL consistently yields comparable generation quality.

## 1 INTRODUCTION

Diffusion models have recently emerged as a powerful paradigm for generating structured, non-Euclidean data, significantly advancing the field of graph generation (Jo et al., 2022; 2024). By defining a forward noising process and learning its reverse dynamics, these models have achieved strong empirical performance in synthesizing complex graph structures.

Graph diffusion models are trained using sample-level denoising objectives such as denoising score matching (DSM) (Vincent, 2011; Song et al., 2021; Jo et al., 2022) or mean squared error (MSE) regression (Jo et al., 2024), which are closely related under Gaussian corruption models and Tweedie's identity (Robbins, 1992; Efron, 2011). Despite their empirical success, this sample-wise training paradigm introduces a fundamental limitation in how learning signals are constructed. Specifically, objectives such as DSM or MSE optimize reconstruction on individual noisy-clean pairs, implicitly encouraging the model to approximate a conditional expectation rather than the full conditional distribution of graphs. While this formulation is statistically optimal under squared-error loss, it inherently reduces rich, multi-modal conditional data structures into a deterministic prediction.

This issue becomes more pronounced due to the combinatorial nature of graph structures, where multiple valid realizations may correspond to the same noisy observation. As a result, the learning process tends to focus on point-wise fidelity while overlooking the consistency of the induced distributional geometry across samples (Jo et al., 2022; 2024). This mismatch between local reconstruction objectives and global structural consistency motivates the need for an explicit regularization that operates at the distribution level rather than solely at the sample level.

To address this limitation, we propose manifold alignment (MAL) as shown in Figure 1, a geometric regularization that complements standard sample-wise denoising objectives. Instead of introducing additional supervision at the individual sample level, MAL focuses on the consistency between the predictive distribution induced by the model and the real data distribution within each batch.

The key insight is that while denoising objectives enforce accurate reconstruction for each noisy instance, they do not explicitly regulate how these local predictions collectively organize into a coherent distribution in the representation space. As a result, even when individual predictions are locally accurate, the aggregated outputs may exhibit inconsistent geometric structures across samples and noise levels.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To bridge this gap, MAL constructs a batch-wise predictive distribution from the network outputs at the current diffusion step and aligns its underlying geometric structure with that of the real data distribution using a non-parametric Fisher divergence. This design introduces an explicit distribution-level constraint that operates orthogonally to the point-wise denoising loss, encouraging global geometric consistency while preserving local reconstruction fidelity.

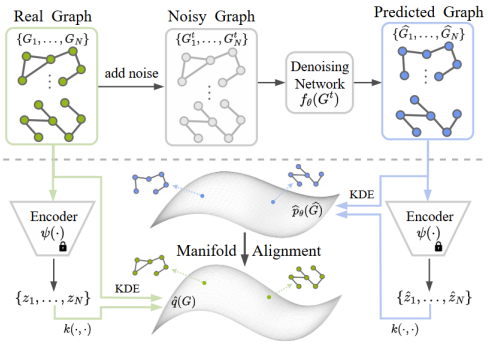



Figure 1: MAL for Graph Diffusion. The top panel illustrates standard graph diffusion and denoising. The bottom panel depicts the MAL regularization framework: real graphs $G$ and predicted graphs $\widehat{G}$ are projected via a frozen pretrained encoder $\psi(\cdot)$ into a shared latent space to compute kernel similarities $k(\cdot,\cdot)$. KDE then constructs the predicted manifold $\widehat{p}_\theta(\widehat{G})$ (top surface) and the real data manifold $\widehat{q}(G)$ (bottom surface). MAL aligns the predicted manifold toward the real data manifold, preserving global geometric consistency.

Geometrically, MAL introduces a batch-level distributional regularization that complements the standard point-wise denoising process. While the model continues to learn local reconstruction mappings, MAL operates on the ensemble of predictions within each mini-batch, enforcing consistency in their induced distributional geometry. This constraint encourages the predictive distribution at each diffusion step to remain aligned with the geometry of the data distribution, rather than being determined solely by independent sample-wise objectives.

From a dynamical perspective, this regularization mitigates the discrepancy between locally accurate denoising predictions and the global structure of the induced sampling distribution. By aligning the predictive geometry with that of the data distribution via a non-parametric divergence, MAL reduces the accumulation of distributional inconsistency across diffusion steps. This leads to more coherent reverse trajectories without modifying the underlying network architecture or the standard denoising objective.

Our main contributions are summarized as follows:

- We identify that standard graph diffusion training is inherently sample-wise, which may lead to inconsistencies in the induced predictive distribution across diffusion steps.
- We propose MAL, a geometric regularization based on non-parametric density estimation that aligns the induced predictive distribution with the data distribution.
- We demonstrate that MAL is a regularizer that can be seamlessly integrated into existing graph diffusion frameworks and consistently improves generation quality.

## 2 Related Work

Graph generative modeling has evolved from early deep autoregressive frameworks (You et al., 2018; Liao et al., 2019; Reiser et al., 2022) to sophisticated latent-variable (Kipf & Welling, 2016; Simonovsky & Komodakis, 2018; De Cao & Kipf, 2018; Jin et al., 2018), flow-based (Zang & Wang, 2020; Shi et al., 2020), and diffusion-based paradigms (Niu et al., 2020; Jo et al., 2022; Vignac et al., 2023; Qin et al., 2025). While autoregressive models capture local patterns through sequential decisions, they often struggle with long-range dependencies and node-ordering sensitivity (Liao et al., 2019). Latent-variable models provide a global perspective but frequently face challenges in reconstructing discrete topologies (Kipf & Welling, 2016; Simonovsky & Komodakis, 2018). Diffusion-based models have recently emerged as a powerful paradigm, synthesizing high-quality graphs by iteratively reversing a stochastic diffusion process (Jo et al., 2022). Despite their success, current frameworks primarily rely on point-wise denoising objectives that optimize each graph as an isolated instance. Such objectives focus on local structural recovery but lack explicit enforcement of geometric consistency across samples within a batch. Consequently, the model fails to capture the intrinsic manifold structure of the distribution, often resulting in structural instability in low-density

regions. To address this, our framework introduces manifold alignment learning, which complements the isolated reconstruction objective by aligning the distributional geometry of the generated samples with the intrinsic geometry of the real data distribution, thereby bridging the gap between local structural recovery and global distributional integrity.

Beyond standard denoising diffusion, diffusion bridge models have become a powerful paradigm for structural transitions between domains by learning stochastic transport processes. Representative frameworks include Diffusion Schrödinger Bridge (DSB) (De Bortoli et al., 2021; Chen et al., 2022), which formulate generative modeling as a stochastic optimal control problem. Concurrently, diffusion mixture approaches have emerged, avoiding the reverse noising process in favor of modeling a mixture of diffusion processes (Peluchetti, 2023; Wu et al., 2022; Liu et al., 2023; Jo & Hwang, 2024). Recently, GruM (Jo et al., 2024) has further advanced this field by introducing a mixture of endpoint-conditioned diffusion bridges, which explicitly models the final graph structure via a weighted graph mixture instead of predicting intermediate denoising scores. Although these bridge-based methods excel at modeling transition trajectories, they are conventionally trained primarily using trajectory-level optimization objectives. While efficient for domain mapping, such optimization lacks explicit regularization for the global manifold geometry of the target distribution. Specifically, when transition trajectories traverse low-density regions of the target manifold, they often result in structural instability or fragmented generation. To address this, our framework introduces manifold alignment learning as a global regularizer, ensuring that learned transition trajectories remain continuously anchored to the intrinsic geometry of the target data distribution, thereby bridging the gap between trajectory optimization and global distributional integrity.

Standard score matching (Hyvärinen, 2005) and its denoising variant (Vincent, 2011) bypass intractable partition function integration by directly estimating log-density gradients. This formulation establishes the underlying mathematical foundation for score-based generative diffusion models, which have evolved from discrete multi-scale Noise Conditional Score Networks (NCSN) (Song & Ermon, 2019) to a unified continuous-time SDE framework (Song et al., 2021). Concurrently, kernel-based non-parametric distribution alignment methods, such as Maximum Mean Discrepancy (MMD) (Gretton et al., 2012; Li et al., 2015; Dziugaite et al., 2015), Kernelized Stein Discrepancy (KSD) (Liu et al., 2016; Chwialkowski et al., 2016), and Stein Variational Gradient Descent (SVGD) (Liu & Wang, 2016), have gained prominence. Under Gaussian kernels, Lai et al. (2026) established an equivalence between drifting fields and smoothed score-mismatch fields, linking single-step drifting to score matching. However, this connection is limited to single-step generation, leaving continuous-time trajectory modeling and geometric regularization largely unexplored. To bridge this gap, we propose MAL. Utilizing batch-level non-parametric density estimation, MAL estimates the local scores of both real and generated distributions, aligning them via Fisher divergence. This alignment essentially constitutes a manifold-consistent gradient constraint, thereby capturing and preserving the global topological manifold.

## 3 Manifold Alignment Learning

### 3.1 Manifold Alignment

For diffusion models (Ho et al., 2020; Song et al., 2021), denoising score matching is equivalent to minimizing squared error-based data $\boldsymbol{x}$-prediction under the Tweedie identity (Robbins, 1992; Efron, 2011). Consequently, both formulations learn the same underlying conditional statistics. For simplicity, we denote the clean graph state as $\boldsymbol{x}$ and its noisy counterpart at timestep $t$ as $\boldsymbol{x}_t$.

The core statistical objective of diffusion-based models is to align the generative distribution $p_\theta(\boldsymbol{x})$ obtained after denoising as closely as possible with the true data distribution $q(\boldsymbol{x})$. Under the continuous-time Gaussian perturbation induced by the forward SDE, the evolution of information along the diffusion trajectory can be characterized through a Fisher information identity (Lyu, 2009). This leads to a decomposition of the KL divergence into a time-integrated Fisher divergence:

$$D_{\mathrm{kl}}\big(q(\boldsymbol{x})\|p_\theta(\boldsymbol{x})\big) = \frac{1}{2}\int_0^T g_t^2 D_f\big(q(\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_t)\big)\mathrm{d}t \tag{1}$$

where $D_f$ denotes the Fisher divergence at time $t$, and $g_t$ is the diffusion coefficient of the forward SDE. Eq. 1 shows that global distribution matching in the clean space is equivalent to continuously matching score discrepancies across all noise levels along the diffusion process.

To optimize this objective of Eq. 1, diffusion models can be viewed as implicitly performing denoising score matching, which connects Fisher divergence minimization to a conditional denoising regression problem. Through the Gaussian perturbation structure and Tweedie identity, the score estimation is equivalently transformed into an $\boldsymbol{x}$-prediction objective. This yields the following equivalence:

$$\frac{1}{2}\int_0^T g_t^2 D_f\big(q(\boldsymbol{x}_t) \parallel p_\theta(\boldsymbol{x}_t)\big)\mathrm{d}t \equiv \int_0^T w_t \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\big[\|\boldsymbol{x}-\widehat{\boldsymbol{x}}\|^2\big]\mathrm{d}t \tag{2}$$

where $w_t$ is a time-dependent weighting parameter (proofs in Appendix A) and $\widehat{\boldsymbol{x}}$ is the denoised prediction at time $t$.

This perspective makes explicit that the optimal denoiser under a squared-error objective is closely related to the posterior mean estimator $\widehat{\boldsymbol{x}} = \mathbb{E}[\boldsymbol{x}|\boldsymbol{x}_t]$ rather than the full posterior distribution $q(\boldsymbol{x}|\boldsymbol{x}_t)$ (Ho et al., 2020; Song et al., 2021; Jo et al., 2022; 2024), which forms the basis of our discussion. This implies that the learned model can be interpreted as a deterministic approximation of the posterior in the form of $p_\theta(\boldsymbol{x}|\boldsymbol{x}_t) = \delta(\boldsymbol{x}-\widehat{\boldsymbol{x}})$, which collapses the inherently multi-modal true posterior $q(\boldsymbol{x}|\boldsymbol{x}_t)$, especially in structured domains such as graphs, into a single-point representation.

This posterior degeneration induces a chain reaction during the reverse sampling process. The single-step reverse transition $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ can be viewed as marginalizing over the unknown clean state: $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \int q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x})p_\theta(\boldsymbol{x}|\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}$ . Theoretically, this ancestral construction assumes that the transition step $\boldsymbol{x}_t \to \boldsymbol{x}_{t-1}$ is guided by the distribution of all potential valid topologies.

However, substituting the deterministic posterior approximation effectively removes integration over the true posterior manifold, leading to a plug-in approximation:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \approx q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\widehat{\boldsymbol{x}}). \tag{3}$$

This reduction reveals a fundamental limitation: the reverse dynamics are conditioned on a single conditional mean $\widehat{\boldsymbol{x}}$, which generally lies outside the valid non-convex structural manifold $\mathcal{M}$. As a result, the reverse trajectory is driven by off-manifold anchors, leading to progressively accumulated structural distortion during sampling.

To quantify this mismatch, we compare the posterior geometry with its degenerate Dirac approximation. This leads to a kernel-density-based perspective, where we align the marginal distribution induced by the predicted clean samples with the empirical data distribution: $\widehat{p}_\theta(\widehat{\boldsymbol{x}}) \approx \widehat{q}(\boldsymbol{x})$ . Based on this observation, we introduce the probability manifold alignment learning (MAL), whose objective is formulated as a score-based distribution alignment:

$$\ell_{\mathrm{mal}} = D_f\big(\widehat{q}(\boldsymbol{x}) \parallel \widehat{p}_\theta(\widehat{\boldsymbol{x}})\big) = \mathbb{E}_{\widehat{q}(\boldsymbol{x})}\left[\|\nabla_{\boldsymbol{x}}\log\widehat{q}(\boldsymbol{x}) - \nabla_{\widehat{\boldsymbol{x}}}\log\widehat{p}_\theta(\widehat{\boldsymbol{x}})\|_2^2\right]. \tag{4}$$

The key insight is that aligning the clean-data distributions naturally induces alignment of their noisy marginals across all diffusion timesteps. The intuition behind this observation is that the clean-state prediction serves as the latent anchor of the reverse diffusion trajectory. Given a noisy state $\boldsymbol{x}_{t+1}$, the reverse transition $q(\boldsymbol{x}_t|\boldsymbol{x}_{t+1},\boldsymbol{x})$ determines the distribution of the preceding state once the clean state $\boldsymbol{x}$ is specified. Importantly, this transition is analytically known and independent of the denoising network. Therefore, replacing the ground-truth clean sample $\boldsymbol{x}$ with the predicted clean sample $\widehat{\boldsymbol{x}}$ directly changes the induced distribution of intermediate states. By aligning the distribution of $\widehat{\boldsymbol{x}}$ with the real data distribution, MAL reduces the distributional discrepancy propagated through the fixed diffusion transitions. Consequently, clean-state alignment provides a principled way to regularize the intermediate noisy distributions implicitly induced by the denoising objective in Eq. 2.

### 3.2 Non-parametric Score Estimation

To estimate the two distributions in Eq. 4, we employ the Parzen kernel density estimation (KDE) (Parzen, 1962; Davis et al., 2011) which approximates densities from finite samples.

Given a mini-batch of $n_b$ real samples $\{\boldsymbol{x}_i\}_{i=1}^{n_b}$, KDE estimates the density at any point $\boldsymbol{x}$ by aggregating the contributions from these batch samples. KDE constructs a smooth density field by aggregating local contributions from these samples. Specifically, a kernel function $k(\cdot,\cdot)$ is centered at each sample to measure its influence on any query point $\boldsymbol{x}$. To ensure differentiability of the resulting density field, we adopt the Gaussian kernel. The resulting density estimate is:

$$\widehat{q}(\boldsymbol{x}) = \frac{1}{n_b}\sum_{i=1}^{n_b} k(\boldsymbol{x},\boldsymbol{x}_i), \tag{5}$$

where the Gaussian kernel is defined as:

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = \frac{1}{(2\pi\sigma^2)^{n_d/2}} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma^2}\right). \tag{6}$$

with $n_d$ denoting the dimensionality of the feature space and $\sigma$ the kernel bandwidth.

Given the empirical density fields $\widehat{q}(\boldsymbol{x})$ defined by Eq. 5, we can analytically derive its corresponding score. Substituting the KDE estimator Eq. 5 into $\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$, the score for the real data distribution can be derived as (details in Appendix A):

$$\nabla_{\boldsymbol{x}} \log \widehat{q}(\boldsymbol{x}) \propto \boldsymbol{\mu}(\boldsymbol{x}) - \boldsymbol{x}\,, \tag{7}$$

where $\boldsymbol{\mu}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)\boldsymbol{x}_i}{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)}$.

Given a mini-batch of $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^{n_b}$, the empirical density of generated samples is approximated as: $\widehat{p}_\theta(\widehat{\boldsymbol{x}}) = \frac{1}{n_b}\sum_{i=1}^{n_b} k(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{x}}_i)$. Similarly, the score of the generated distribution is given by

$$\nabla_{\widehat{\boldsymbol{x}}} \log \widehat{p}_\theta(\widehat{\boldsymbol{x}}) \propto \boldsymbol{\mu}(\widehat{\boldsymbol{x}}) - \widehat{\boldsymbol{x}}\,. \tag{8}$$

### 3.3 Beyond Euclidean Graph Similarity

KDE relies on a notion of distance to define similarity between samples. In its standard form, this is instantiated via Euclidean distance, i.e., $k(\boldsymbol{x}, \boldsymbol{x}_i) \propto -\|\boldsymbol{x} - \boldsymbol{x}_i\|^2$, which implicitly assumes that the underlying data space is equipped with an isotropic Euclidean geometry. As a result, directly measuring $\|\boldsymbol{x} - \boldsymbol{x}_i\|^2$ provides a poor proxy for graph-level similarity, leading KDE to construct density fields over a geometry that does not reflect the true data manifold.

However, in graph domains, the notion of similarity is inherently defined by high-level structural and semantic equivalence rather than raw feature proximity. Two graphs may be semantically identical while being far apart in the Euclidean embedding of node-level features, or conversely, appear close in Euclidean space while being structurally inconsistent.

Thus, a meaningful kernel should be defined not in the raw feature space, but in a representation space where distances correspond to semantic and structural similarity between graphs. We employ a pretrained graph encoder $\psi(\cdot)$ to extract robust semantic representations for both real and generated graphs, as depicted in the lower panel of Figure 1. By computing similarities in this latent space, we enhance the precision of the KDE-based density estimation. Specifically, for a graph $G = (X, A)$ with node feature matrix $X \in \mathbb{R}^{N \times d_v}$ and adjacency matrix $A \in \mathbb{R}^{N \times N}$, we obtain its semantic representation $\boldsymbol{z} = \psi(G)$, and compute the refined similarity weights as: $k(\boldsymbol{z}, \boldsymbol{z}_i)$ and $k(\widehat{\boldsymbol{z}}, \widehat{\boldsymbol{z}}_i)$.

Substituting these semantic-aware similarities into Eq. 7 and Eq. 8 effectively aligns the scores within the latent semantic manifold. This projection strategy provides two-fold benefits: first, it ensures that the learned scores are aligned with the high-level manifold structure of the graph data; second, it filters out irrelevant structural noise, enabling the KDE to focus on semantically meaningful neighborhoods.

### 3.4 MAL Regularization for Graph Generation

The overall framework of our proposed MAL is illustrated in Figure 1. To provide a comprehensive overview of the proposed training procedure, we delineate the complete manifold alignment learning process in Algorithm 1. To regularize the graph generative process within the data manifold, we introduce MAL, a Fisher divergence-based consistency objective, Eq. 4, which aligns the score fields of the real and generated distributions. Eq. 4 enforces agreement between the local geometric structures induced by the two distributions, thereby encouraging generated samples to remain in high-density regions of the data manifold. This geometric regularization ensures that the generated graphs are continuously guided toward the high-density regions of the data manifold, effectively preventing the generation of out-of-distribution structures.

**Algorithm 1** MAL Training Algorithm.

**Input:** A training set $\mathcal{D}$, pretrained $\psi(\cdot)$, $\lambda$, $\sigma$.
**Initialization:** Model $\theta$.
**for** $t \sim \text{uniform}(0, T)$ **do**
  $\{G_j\}_{j=1}^{n_b} \sim \mathcal{D}$ where $G_j = (X_j, A_j)$;
  **for** $j \in [1, n_b]$ **do**
    $X_t \sim q(X_t|X_j; \boldsymbol{\epsilon}_x), \boldsymbol{\epsilon}_x \sim \mathcal{N}(0, I)$;
    $A_t \sim q(A_t|A_j; \boldsymbol{\epsilon}_a), \boldsymbol{\epsilon}_a \sim \mathcal{N}(0, I)$;
    $\widehat{G}_j = f_\theta(G_t, t)$ where $G_t = (X_t, A_t)$;
    $\boldsymbol{z}_j = \psi(G_j), \quad \widehat{\boldsymbol{z}}_j = \psi(\widehat{G}_j)$;
  **end for**
  Evaluate Gaussian kernel in Eq. 6 in embedding space, yielding $k(\boldsymbol{z}_j, \boldsymbol{z}_i)$ and $k(\widehat{\boldsymbol{z}}_j, \widehat{\boldsymbol{z}}_i)$.
  Calculate scores by Eq. 7 and Eq. 8;
  Calculate $\mathcal{L}_{\text{mal}}$ by Eq. 9;
  Calculate $\mathcal{L}_{\text{mse}}$ by Eq. 10;
  $\mathcal{L} = \mathcal{L}_{\text{mse}} + \lambda \mathcal{L}_{\text{mal}}$;
  Update $\theta$ by minimizing $\mathcal{L}$;
**end for**
**Output:** $\theta^\star = \theta$.

Specifically, we formulate the score-field regularization loss $\mathcal{L}_{\text{mal}}$ as the mean squared error between the empirical scores of the real and generated samples:

$$\mathcal{L}_{\text{mal}} = \sum_{j=1}^{n_b} \left\| (\boldsymbol{\mu}(X_j) - X_j) - \left( \text{sg}\left( \boldsymbol{\mu}(\widehat{X}_j) \right) - \widehat{X}_j \right) \right\|^2 + \left\| (\boldsymbol{\mu}(\boldsymbol{A}_j) - A_j) - \left( \text{sg}\left( \boldsymbol{\mu}(\widehat{A}_j) \right) - \widehat{A}_j \right) \right\|^2 . \tag{9}$$

where the stop-gradient operator $\text{sg}(\cdot)$ plays a critical role in stabilizing the optimization process (Chen & He, 2021).To combine this manifold-level geometric constraint with point-wise generative performance, we seamlessly integrate it into the standard diffusion framework. The reconstruction loss is defined as:

$$\mathcal{L}_{\text{mse}} = \sum_{j=1}^{n_b} \left[ \|X_j - \widehat{X}_j\|^2 + \|A_j - \widehat{A}_j\|^2 \right] \tag{10}$$

where $\widehat{X}_j$ and $\widehat{A}_j$ denote the denoised predictions at time $t$. While $\mathcal{L}_{\text{mse}}$ ensures local point-wise fidelity, the regularization loss $\mathcal{L}_{\text{mal}}$ ensures topological accuracy by aligning the global score-field. The overall training loss is:

$$\mathcal{L} = \mathcal{L}_{\text{mse}} + \lambda \mathcal{L}_{\text{mal}} \tag{11}$$

where $\lambda$ balances denoising fidelity and manifold-level geometric regularization. This synergy allows the model to reconstruct local graph patterns while injecting high-order structural information, ensuring robust graph generation even in low-density regions of the data space.

## 4 EXPERIMENT

Before conducting benchmark evaluations, we first verify the plug-and-play nature of MAL by integrating it into four distinct representative graph diffusion backbones: GDSS (Jo et al., 2022), GruM (Jo et al., 2024), DiGress (Vignac et al., 2023), and DeFoG (Qin et al., 2025). We evaluate the generation quality on the molecular QM9 dataset with ($\surd$) and without ($\times$) our proposed MAL regularization. As summarized in Table 1, integrating MAL consistently enhances the generation quality across all evaluated backbones, validating its plug-and-play nature. First, MAL universally boosts the chemical validity of the generated molecules across all models, with GDSS improving from 97.19% to 98.02% and GruM achieving a near-perfect 99.93% validity. Second, consistent reductions in `FCD` and `NSPDK` across all backbones demonstrate that MAL effectively pulls generated samples toward the real-data manifold. Lastly, steady improvements in scaffold similarity (`Scaf.`) confirm that MAL successfully preserves macro-structural semantics and global geometric properties.

While MAL delivers solid performance gains on all baselines, its integration with GruM yields the most competitive quality (as detailed in subsequent sections). Driven by this observation, we select GruM as our primary backbone for the benchmark evaluations on both molecular graph generation (§4.2) and general graph generation (§4.3) to showcase the potential of our approach.

Table 1: Generalizability and plug-and-play evaluation of MAL across various graph diffusion backbones on QM9. ✓ and × indicate the presence and absence of MAL, respectively.

| Backbone | MAL | Valid ↑ | FCD ↓ | NSPDK ↓ | Scaf. ↑ |
|---|---|---|---|---|---|
| GDSS | × | 97.19 | 2.9918 | 0.0037 | 0.6936 |
| | √ | 98.02 | 2.7696 | 0.0028 | 0.6939 |
| GruM | × | 99.86 | 0.1240 | 0.0004 | 0.9357 |
| | √ | 99.93 | 0.0970 | 0.0002 | 0.9372 |
| DiGress | × | 98.89 | 0.1146 | 0.0003 | 0.9472 |
| | √ | 99.08 | 0.0922 | 0.0003 | 0.9485 |
| DeFoG | × | 96.96 | 0.1102 | 0.0003 | 0.9500 |
| | √ | 97.32 | 0.1036 | 0.0003 | 0.9478 |

### 4.1 Experimental Setting

To ensure the latent space captures meaningful structural and semantic priors, we pretrain the encoder $\psi(\cdot)$ from scratch using domain-specific self-supervised frameworks. Specifically, for molecular graphs, we employ MolCLR (Wang et al., 2022) based on a GINE architecture, which extracts robust chemical and structural invariants through contrastive learning. For general graphs, we adopt GraphMAE (Hou et al., 2022) with a GIN backbone, which we pretrain via masked graph autoencoding to capture complex, high-order topological dependencies. These pretrained encoders project both real and generated graphs into a structured latent space, ensuring that similarity computations remain semantically informed. Full pretraining objectives, backbone integration setups, hyperparameter configurations ($\lambda, \sigma, n_b$), and sensitivity analyses are detailed in Appendix B.

### 4.2 Molecule Generation

#### 4.2.1 Datasets and Metrics.

We evaluate our method on two standard molecular benchmarks: QM9 (Ramakrishnan et al., 2014) and ZINC250k (Irwin et al., 2012). Following standard evaluation protocols (Shi et al., 2020; Preuer et al., 2018), we measure generation quality via chemical Validity (`Valid` ↑), Fréchet ChemNet Distance (`FCD` ↓), Neighborhood Subgraph Pairwise Distance Kernel MMD (`NSPDK` ↓), and Scaffold similarity (`Scaf.` ↑). Dataset and metric details are in Appendix B.1.

#### 4.2.2 Baselines.

To evaluate generative performance, we compare MAL against representative baselines across diverse generative paradigms: (i) flow-based and autoregressive models, including MoFlow (Zang & Wang, 2020), GraphAF (Shi et al., 2020), and GraphDF (Luo et al., 2021); (ii) score-based frameworks, including EDP-GNN (Niu et al., 2020) and GDSS (Jo et al., 2022); and (iii) advanced continuous and discrete graph diffusion variants, including DiGress (Vignac et al., 2023), GraphARM (Kong et al., 2023), GruM (Jo et al., 2024), DeFoG (Qin et al., 2025), and HOG-Diff (Huang & Birdal, 2026).

#### 4.2.3 Results.

The results on QM9 and ZINC250k are summarized in Table 2. Specifically, MAL achieves competitive performance across all evaluation metrics on QM9, obtaining a chemical validity of 99.93% without post-hoc correction, alongside a lower `FCD` of 0.0970 and `NSPDK` of 0.0002. These gains indicate that the generated molecules align closely with the target distribution in both chemical representation and structural topology. On the more complex ZINC250k dataset, MAL maintains robust performance, securing a validity of 99.75%, an `FCD` of 1.5800, and an improved `Scaf.` of 0.5877. These findings suggest that integrating manifold-level consistency regularization effectively mitigates topological blurring, thereby guiding the reverse diffusion trajectory toward valid structural regions of the graph space.

Table 2: The comparison of MAL with baselines on QM9 and ZINC.

| | QM9 | | | | ZINC | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | Valid↑ | FCD↓ | NSPDK↓ | Scaf.↑ | Valid↑ | FCD↓ | NSPDK↓ | Scaf.↑ |
| MoFlow | 91.36 | 4.4670 | 0.0169 | 0.1447 | 63.11 | 20.931 | 0.0455 | 0.0133 |
| GraphAF | 74.43 | 5.6250 | 0.0267 | 0.3046 | 68.47 | 16.023 | 0.0442 | 0.0672 |
| GraphDF | 93.88 | 10.928 | 0.0046 | 0.3778 | 90.61 | 33.546 | 0.1770 | 0.0000 |
| EDP-GNN | 47.52 | 2.6800 | 0.0046 | 0.2302 | 82.97 | 16.737 | 0.0485 | 0.0000 |
| GDSS | 97.19 | 2.9918 | 0.0037 | 0.6936 | 97.01 | 14.656 | 0.0195 | 0.0467 |
| DiGress | 98.89 | 0.1146 | 0.0003 | 0.9472 | 94.99 | 3.4820 | 0.0021 | 0.4163 |
| GraphARM | 90.25 | 1.2200 | 0.0020 | – | 88.23 | 16.260 | 0.0550 | – |
| GruM | 99.86 | 0.1240 | 0.0004 | 0.9357 | 98.34 | 2.2070 | 0.0016 | 0.5278 |
| DeFoG | 96.96 | 0.1102 | 0.0003 | 0.9500 | 96.65 | 2.1230 | 0.0022 | 0.4245 |
| HOG-Diff | 98.74 | 0.1720 | 0.0003 | – | 98.56 | 1.6330 | 0.0010 | – |
| **MAL** (Ours) | 99.93 | 0.0970 | 0.0002 | 0.9372 | 99.75 | 1.5800 | 0.0013 | 0.5877 |

Table 3: The comparison of MAL with baselines on Planar and SBM.

| | Planar | | | | SBM | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | Deg.↓ | Clus.↓ | Orbit↓ | Spec.↓ | Deg.↓ | Clus.↓ | Orbit↓ | Spec.↓ |
| GraphRNN | 0.0049 | 0.2779 | 1.2543 | 0.0459 | 0.0055 | 0.0584 | 0.0785 | 0.0065 |
| GRAN | 0.0007 | 0.0426 | 0.0009 | 0.0075 | 0.0113 | 0.0553 | 0.0540 | 0.0054 |
| SPECTRE | 0.0005 | 0.0785 | 0.0012 | 0.0112 | 0.0015 | 0.0521 | 0.0412 | 0.0056 |
| EDP-GNN | 0.0044 | 0.3187 | 1.4986 | 0.0813 | 0.0011 | 0.0552 | 0.0520 | 0.0070 |
| GDSS | 0.0041 | 0.2676 | 0.1720 | 0.0370 | 0.0212 | 0.0646 | 0.0894 | 0.0128 |
| ConGress | 0.0048 | 0.2728 | 1.2950 | 0.0418 | 0.0273 | 0.1029 | 0.1148 | – |
| GruM | 0.0007 | 0.0489 | 0.0048 | 0.0075 | 0.0006 | 0.0498 | 0.0455 | 0.0051 |
| DeFoG | 0.0005 | 0.0501 | 0.0006 | 0.0072 | 0.0006 | 0.0517 | 0.0556 | 0.0054 |
| HOG-Diff | – | – | – | – | 0.0028 | 0.0500 | 0.0428 | 0.0043 |
| **MAL** (Ours) | 0.0003 | 0.0360 | 0.0029 | 0.0067 | 0.0000 | 0.0491 | 0.0320 | 0.0034 |

## 4.3 General Graph Generation

### 4.3.1 Datasets and Metrics.

We evaluate our method on four general graph benchmarks: Planar (Kipf & Welling, 2017), SBM (Holland et al., 1983), Enzymes (Schomburg et al., 2004), and Grid (You et al., 2018). Following standard protocols (You et al., 2018; Martinkus et al., 2022), we measure structural fidelity via Maximum Mean Discrepancy (MMD) across degree distribution (`Deg.` ↓), clustering coefficient (`Clus.` ↓), 4-node orbit counts (`Orbit` ↓), and Laplacian spectrum (`Spec.` ↓). See Appendix B.1 for details.

### 4.3.2 Baselines.

To evaluate MAL, we compare it against baselines across graph generative paradigms. These include autoregressive models GraphRNN (You et al., 2018) and GRAN (Liao et al., 2019), the one-shot variational autoencoder GraphVAE, and SPECTRE (Martinkus et al., 2022). The remaining generative and diffusion frameworks, including GraphAF, GraphDF, EDP-GNN, GDSS, ConGress (Vignac et al., 2023), GruM, DeFoG, and HOG-Diff, have been introduced in preceding sections.

Table 4: The comparison of MAL with baselines on Enzymes and Grid.

| Method | Enzymes | | | | Grid | | | |
|---|---|---|---|---|---|---|---|---|
| | Deg.↓ | Clus.↓ | Orbit↓ | Avg.↓ | Deg.↓ | Clus.↓ | Orbit↓ | Avg.↓ |
| GraphRNN | 0.0170 | 0.0620 | 0.0460 | 0.0417 | 0.0640 | 0.0430 | 0.0210 | 0.0427 |
| GraphAF | 1.6690 | 1.2830 | 0.2660 | 1.0727 | – | – | – | – |
| GraphDF | 1.5030 | 1.0610 | 0.2020 | 0.9220 | – | – | – | – |
| GraphVAE | 1.3690 | 0.6290 | 0.1910 | 0.7297 | 1.6190 | 0.0000 | 0.9190 | 0.8460 |
| EDP-GNN | 0.0230 | 0.2680 | 0.0820 | 0.1243 | 0.4550 | 0.2380 | 0.3280 | 0.3403 |
| GDSS | 0.0260 | 0.0610 | 0.0090 | 0.0320 | 0.1110 | 0.0050 | 0.0700 | 0.0620 |
| GruM | 0.0007 | 0.0488 | 0.0092 | 0.0196 | 0.0001 | 0.4513 | 0.0164 | 0.1559 |
| HOG-Diff | 0.0110 | 0.0610 | 0.0070 | 0.0263 | – | – | – | – |
| **MAL** (Ours) | 0.0013 | 0.0453 | 0.0058 | 0.0175 | 0.0000 | 0.2848 | 0.0113 | 0.0987 |

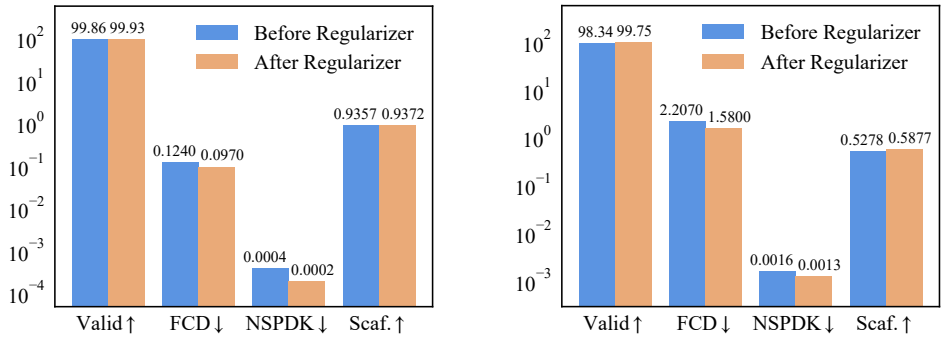



Figure 2: Ablation analysis of the proposed regularizer on QM9 (left) and ZINC (right) datasets.

#### 4.3.3 Results.

As summarized in Tables 3 to 4, MAL achieves competitive generation quality across multiple structural metrics. Specifically, on the Planar and SBM datasets, MAL exhibits favorable performance, obtaining lower discrepancy values in degree distribution (0.0003 on Planar, 0.0000 on SBM) and clustering coefficient (0.0360 on Planar, 0.0491 on SBM). For the real-world Enzymes dataset, MAL shows strong structural alignment, achieving a competitive average score of 0.0175, which outperforms advanced diffusion baselines like GruM and HOG-Diff. On the regular Grid dataset, MAL achieves state-of-the-art performance in degree (0.0000) and orbit count (0.0113) while mitigating GruM's clustering error, effectively preserving grid-like regularities.

### 4.4 Ablation Study

Ablation on GruM (Figure 2) shows consistent gains on QM9, with chemical validity rising to 99.93%, `FCD` dropping to 0.0970, and `NSPDK` halved to 0.0002. On the larger ZINC dataset, MAL similarly boosts validity from 98.34% to 99.75%, reduces `FCD` from 2.2070 to 1.5800, and yields a 6.0% increase in scaffold similarity, demonstrating that manifold constraints effectively prevent trajectory drift (visuals in Appendix C).

## 5 Conclusion

We propose MAL, a plug-and-play geometric regularizer that aligns batch-level predictive distributions with the data manifold via non-parametric Fisher divergence. Bridging sample-wise denoising and global topological consistency across continuous and discrete backbones, MAL achieves superior generation quality on molecular and general graph benchmarks.

## AI USE STATEMENT

Large language models and AI-assisted tools were employed solely for minor text polishing, grammatical refinement, and assistance with code formatting. All foundational research ideas, theoretical derivations, experimental protocols, and scientific analyses were conceived and conducted independently by the authors. The authors have meticulously reviewed and verified all content and assume full responsibility for the entirety of this manuscript.

## ETHICS STATEMENT

This research relies exclusively on standard, publicly accessible benchmark datasets and open-source packages, used in strict adherence to their respective licensing agreements and intended purposes. This work does not involve human subjects, animal testing, personally identifiable information (PII), or private data. We foresee no direct discriminatory, unethical, or harmful applications resulting from our methodological advancements, and we confirm that this study fully complies with the applicable Code of Ethics.

## REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our findings, comprehensive descriptions of the network architectures, implementation details, baseline configurations, and training hyperparameters are thoroughly documented in Section 4 and Appendix B. The complete source code, data preprocessing pipelines, trained model checkpoints, and evaluation scripts will be made publicly available in an open-source repository upon publication.

## REFERENCES

Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. In *ICLR*, 2022.

Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. In *CVPR*, pp. 15750–15758, 2021.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In *ICML*, pp. 2606–2615, 2016.

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *ICML*, pp. 255–262, 2010.

Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. Remarks on some nonparametric estimates of a density function. In *Selected Works of Murray Rosenblatt*, pp. 95–100, 2011.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *NeurIPS*, volume 34, pp. 17695–17709, 2021.

Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. In *ICML Workshop*, 2018.

Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, volume 31, pp. 258–267, 2015.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pp. 6840–6851, 2020.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-supervised masked graph autoencoders. In *KDD*, pp. 594–604, 2022.

Yiming Huang and Tolga Birdal. Hog-diff: Higher-order guided diffusion for graph generation. In *ICLR*, volume 14, 2026.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

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.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *ICML*, pp. 2323–2332, 2018.

Jaehyeong Jo and Sung Ju Hwang. Generative modeling on manifolds through mixture of Riemannian diffusion processes. In *ICML*, volume 41, pp. 22348–22370, 2024.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *ICML*, pp. 10362–10383, 2022.

Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. In *ICML*, volume 41, pp. 22371–22405, 2024.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In *ICML*, pp. 17391–17408, 2023.

Chieh-Hsin Lai, Bac Nguyen, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, Stefano Ermon, and Molei Tao. A unified view of score-based and drifting models. *arXiv preprint arXiv:2603.07514*, 2026.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, pp. 1718–1727, 2015.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *NeurIPS*, volume 32, 2019.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NeurIPS*, volume 29, 2016.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *ICML*, pp. 276–284, 2016.

Xingchao Liu, Lemeng Wu, Mao Ye, et al. Learning diffusion bridges on constrained domains. In *ICLR*, 2023.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. GraphDF: A discrete flow model for molecular graph generation. In *ICML*, volume 38, pp. 7192–7203, 2021.

Siwei Lyu. Interpretation and generalization of score matching. In *UAI*, volume 25, pp. 359–366, 2009.

Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. Spectre: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In *ICML*, pp. 15159–15179, 2022.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *AISTATS*, pp. 4474–4484, 2020.

Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

Stefano Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter 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.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. DeFoG: Discrete flow matching for graph generation. In *ICML*, pp. 50269–50326, 2025.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. Graph neural networks for materials science and chemistry. *Communications Materials*, 3:93, 2022.

Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and Basic Theory*, pp. 388–394. 1992.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. BRENDA, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl_1):D431–D433, 2004.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: a flow-based autoregressive model for molecular graph generation. In *ICLR*, volume 8, 2020.

Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. In *NeurIPS*, pp. 412–422, 2018.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, volume 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *ICLR*, 2023.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, pp. 279–287, 2022.

Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. In *NeurIPS*, volume 35, pp. 36533–36545, 2022.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *ICML*, pp. 5708–5717, 2018.

Chengxi Zang and Fei Wang. MoFlow: An invertible flow model for generating molecular graphs. In *KDD*, pp. 617–626, 2020.

# Appendix

## A Detailed Mathematical Derivations

In this section, we present the comprehensive step-by-step mathematical derivations for the key equations and theoretical formulations presented in Section 3 of the main manuscript.

### A.1 Proof of Equation 1: KL Divergence Decomposition

We establish the theoretical proof for this time-integrated Fisher information identity, which connects the global KL divergence between the clean data distribution $q(\boldsymbol{x})$ and model distribution $p_\theta(\boldsymbol{x})$ to the cumulative Fisher divergence along the continuous SDE trajectory.

**Fokker-Planck Dynamics.** Consider a continuous-time diffusion process defined by the forward Stochastic Differential Equation (SDE):

$$\mathrm{d}\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)\mathrm{d}t + g_t \mathrm{d}\boldsymbol{w}_t \tag{12}$$

where $\boldsymbol{f}(\boldsymbol{x}_t, t)$ represents the drift coefficient, $g_t$ is the diffusion coefficient, and $\boldsymbol{w}_t$ is the standard Brownian motion.

The marginal probability densities of the true data $q(\boldsymbol{x}_t)$ and the model distribution $p_\theta(\boldsymbol{x}_t)$ evolve over time according to their respective Fokker-Planck equations (also known as Kolmogorov forward equations):

$$\frac{\partial q(\boldsymbol{x}_t)}{\partial t} = -\nabla_{\boldsymbol{x}_t} \cdot (\boldsymbol{f}(\boldsymbol{x}_t, t) q(\boldsymbol{x}_t)) + \frac{1}{2} g_t^2 \Delta_{\boldsymbol{x}_t} q(\boldsymbol{x}_t) \tag{13}$$

$$\frac{\partial p_\theta(\boldsymbol{x}_t)}{\partial t} = -\nabla_{\boldsymbol{x}_t} \cdot (\boldsymbol{f}(\boldsymbol{x}_t, t) p_\theta(\boldsymbol{x}_t)) + \frac{1}{2} g_t^2 \Delta_{\boldsymbol{x}_t} p_\theta(\boldsymbol{x}_t) \tag{14}$$

where $\Delta_{\boldsymbol{x}_t} = \nabla_{\boldsymbol{x}_t} \cdot \nabla_{\boldsymbol{x}_t}$ represents the Laplace operator.

**Time Derivative of KL Divergence.** The Kullback-Leibler (KL) divergence between $q(\boldsymbol{x}_t)$ and $p_\theta(\boldsymbol{x}_t)$ at any intermediate timestep $t$ is defined as:

$$D_{\mathrm{kl}}\big(q(\boldsymbol{x}_t) \| p_\theta(\boldsymbol{x}_t)\big) = \int q(\boldsymbol{x}_t) \log \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \mathrm{d}\boldsymbol{x}_t \tag{15}$$

Differentiating both sides with respect to time $t$ using the Leibniz integral rule yields:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} D_{\mathrm{kl}}\big(q(\boldsymbol{x}_t) \| p_\theta(\boldsymbol{x}_t)\big) &= \int \frac{\partial q(\boldsymbol{x}_t)}{\partial t} \log \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \mathrm{d}\boldsymbol{x}_t + \int q(\boldsymbol{x}_t) \frac{\partial}{\partial t} \left(\log q(\boldsymbol{x}_t) - \log p_\theta(\boldsymbol{x}_t)\right) \mathrm{d}\boldsymbol{x}_t \\
&= \int \frac{\partial q(\boldsymbol{x}_t)}{\partial t} \log \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \mathrm{d}\boldsymbol{x}_t + \int \left( \frac{\partial q(\boldsymbol{x}_t)}{\partial t} - \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \frac{\partial p_\theta(\boldsymbol{x}_t)}{\partial t} \right) \mathrm{d}\boldsymbol{x}_t
\end{aligned} \tag{16}$$

Since the total probability is conserved (i.e., $\int q(\boldsymbol{x}_t) \mathrm{d}\boldsymbol{x}_t = 1$), its derivative over time vanishes, $\int \frac{\partial q(\boldsymbol{x}_t)}{\partial t} \mathrm{d}\boldsymbol{x}_t = 0$. Therefore, the derivative simplifies to:

$$\frac{\mathrm{d}}{\mathrm{d}t} D_{\mathrm{kl}}\big(q(\boldsymbol{x}_t) \| p_\theta(\boldsymbol{x}_t)\big) = \int \frac{\partial q(\boldsymbol{x}_t)}{\partial t} \log \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \mathrm{d}\boldsymbol{x}_t - \int \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \frac{\partial p_\theta(\boldsymbol{x}_t)}{\partial t} \mathrm{d}\boldsymbol{x}_t \tag{17}$$

**Integration by Parts.** We substitute the Fokker-Planck equations into Eq. 17. By applying Green's first identity (equivalently, integration by parts in the multidimensional continuous domain) and assuming that the probability densities and their gradients vanish at the boundaries of the domain (i.e., $\lim_{\|\boldsymbol{x}_t\| \to \infty} q(\boldsymbol{x}_t) = 0$), the drift contributions vanish after applying integration by parts. This yields:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} D_{\mathrm{kl}}\big(q(\boldsymbol{x}_t) \| p_\theta(\boldsymbol{x}_t)\big) &= \int \left( \frac{1}{2} g_t^2 \Delta_{\boldsymbol{x}_t} q(\boldsymbol{x}_t) \right) \log \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \mathrm{d}\boldsymbol{x}_t - \int \frac{q(\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_t)} \left( \frac{1}{2} g_t^2 \Delta_{\boldsymbol{x}_t} p_\theta(\boldsymbol{x}_t) \right) \mathrm{d}\boldsymbol{x}_t \\
&= -\frac{1}{2} g_t^2 \int q(\boldsymbol{x}_t) \left\| \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log p_\theta(\boldsymbol{x}_t) \right\|_2^2 \mathrm{d}\boldsymbol{x}_t
\end{aligned} \tag{18}$$

This formulation is known as the Generalized De Bruijn's Identity:

$$\frac{\mathrm{d}}{\mathrm{d}t} D_{\mathrm{kl}}\big(q(\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_t)\big) = -\frac{1}{2} g_t^2 D_f\big(q(\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_t)\big) \tag{19}$$

where $D_{\mathrm{F}}$ represents the unconditioned Fisher divergence at timestep $t$.

**Boundary Conditions and Integration.** We perform integration on both sides of Eq. 19 over the entire diffusion trajectory $t \in [0, T]$:

$$\int_0^T \frac{\mathrm{d}}{\mathrm{d}t} D_{\mathrm{kl}}\big(q(\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_t)\big)\mathrm{d}t = -\frac{1}{2}\int_0^T g_t^2 D_f\big(q(\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_t)\big)\mathrm{d}t \tag{20}$$

Applying the Fundamental Theorem of Calculus, the left-hand side reduces to the boundary difference:

$$D_{\mathrm{kl}}\big(q(\boldsymbol{x}_T)\|p_\theta(\boldsymbol{x}_T)\big) - D_{\mathrm{kl}}\big(q(\boldsymbol{x})\|p_\theta(\boldsymbol{x})\big) = -\frac{1}{2}\int_0^T g_t^2 D_f\big(q(\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_t)\big)\mathrm{d}t \tag{21}$$

At this point, we introduce the terminal boundary condition of diffusion models: at time $t = T$, the data distribution is completely submerged by intense Gaussian noise under the forward perturbation process. Consequently, both the true perturbed marginal distribution $q(\boldsymbol{x}_T)$ and the trainable prior distribution $p_\theta(\boldsymbol{x}_T)$ asymptotically converge to the identical parameter-free standard Gaussian prior distribution, i.e., $q(\boldsymbol{x}_T) \approx p_\theta(\boldsymbol{x}_T) \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Consequently, their KL divergence at $t = T$ vanishes:

$$D_{\mathrm{kl}}\big(q(\boldsymbol{x}_T)\|p_\theta(\boldsymbol{x}_T)\big) \approx 0 \tag{22}$$

Substituting this terminal boundary condition back into Eq. 21 and multiplying both sides by $-1$, we obtain the final time-integrated Fisher information identity (Equation 1 of the main manuscript):

$$D_{\mathrm{kl}}\big(q(\boldsymbol{x})\|p_\theta(\boldsymbol{x})\big) = \frac{1}{2}\int_0^T g_t^2 D_f\big(q(\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_t)\big)\mathrm{d}t \tag{1}$$

This completes the formal proof of Equation 1.

### A.2 Proof of Equation 2: Equivalence of Fisher Divergence and Denoising Loss

We present the step-by-step mathematical proof demonstrating that the time-integrated Fisher divergence objective is equivalent to the sample-wise conditional denoising mean squared error (MSE) objective used in standard diffusion training.

**Denoising Score Matching Theorem.** The unconditioned Fisher divergence at a specific timestep $t$ is defined as:

$$D_f(q(\boldsymbol{x}_t) \parallel p_\theta(\boldsymbol{x}_t)) = \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\|\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\|_2^2\right] \tag{23}$$

where $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t}\log p_\theta(\boldsymbol{x}_t)$ represents the score network. Since the true marginal data distribution $q(\boldsymbol{x}_t) = \int q(\boldsymbol{x})q(\boldsymbol{x}_t|\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ involves an intractable high-dimensional integral, its score $\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t)$ cannot be computed directly. The Denoising Score Matching (DSM) theorem Vincent (2011) establishes that, up to a constant independent of the parameters $\theta$, the unconditioned score matching objective is equivalent to the conditional score matching objective:

$$\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\|\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\|_2^2\right] \equiv \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\left[\|\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\|_2^2\right] \tag{24}$$

To prove this equivalence, we expand the quadratic norm of the right-hand side of Eq. 24 and focus on the cross term containing the optimization parameter $\theta$:

$$\begin{aligned}
&\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\left[\|\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\|_2^2\right] \\
&= \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\left[\|\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x})\|_2^2\right] - 2\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\left[\boldsymbol{s}_\theta^\top(\boldsymbol{x}_t, t)\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x})\right] + \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\|_2^2\right]
\end{aligned} \tag{25}$$

where the algebraic identity $\|\boldsymbol{a}-\boldsymbol{b}\|_2^2 = \|\boldsymbol{a}\|_2^2 - 2\boldsymbol{a}^\top\boldsymbol{b} + \|\boldsymbol{b}\|_2^2$ is applied.

Since the first term in Eq. 25 is independent of the parameters $\theta$, and the third term $\mathbb{E}_{q(\boldsymbol{x}_t)}[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\|_2^2]$ is identical to the corresponding quadratic term in the expansion of the unconditioned objective (the LHS of Eq. 24), the optimization trajectories of both objectives are determined entirely by their respective cross terms. Writing the cross term of the conditional objective in its integral representation yields:

$$-2\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\left[\boldsymbol{s}_\theta^\top(\boldsymbol{x}_t,t)\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x})\right] = -2\iint q(\boldsymbol{x},\boldsymbol{x}_t)\boldsymbol{s}_\theta^\top(\boldsymbol{x}_t,t)\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}_t \quad (26)$$

Using the relation $q(\boldsymbol{x},\boldsymbol{x}_t) = q(\boldsymbol{x})q(\boldsymbol{x}_t|\boldsymbol{x})$ and the identity $\nabla\log q = \frac{\nabla q}{q}$, this joint-distribution cross term can be simplified by canceling out $q(\boldsymbol{x}_t|\boldsymbol{x})$ in the denominator:

$$-2\int \boldsymbol{s}_\theta^\top(\boldsymbol{x}_t,t)\left(\int q(\boldsymbol{x})\nabla_{\boldsymbol{x}_t}q(\boldsymbol{x}_t|\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)\mathrm{d}\boldsymbol{x}_t \quad (27)$$

By Leibniz's integral rule, since the boundaries of the integration domain are independent of $t$, we can swap the order of integration and differentiation inside the parentheses:

$$\int q(\boldsymbol{x})\nabla_{\boldsymbol{x}_t}q(\boldsymbol{x}_t|\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \nabla_{\boldsymbol{x}_t}\int q(\boldsymbol{x})q(\boldsymbol{x}_t|\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \nabla_{\boldsymbol{x}_t}q(\boldsymbol{x}_t) \quad (28)$$

Substituting this result back into the simplified cross term, we arrive at:

$$-2\int q(\boldsymbol{x}_t)\boldsymbol{s}_\theta^\top(\boldsymbol{x}_t,t)\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t = -2\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\boldsymbol{s}_\theta^\top(\boldsymbol{x}_t,t)\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t)\right] \quad (29)$$

which is mathematically identical to the cross term obtained by expanding the unconditioned Fisher divergence (the LHS of Eq. 24). Since all non-constant terms with respect to $\theta$ match exactly, the equivalence between explicit score matching and denoising score matching is established.

**Conditional Transition Probability and Score.** In diffusion models, the forward perturbation process from the clean graph state $\boldsymbol{x}$ to the noisy state $\boldsymbol{x}_t$ is typically modeled as a Gaussian transition:

$$\boldsymbol{x}_t = \alpha_t\boldsymbol{x} + \sigma_t\epsilon, \quad \epsilon\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}) \quad (30)$$

where $\alpha_t$ and $\sigma_t$ represent the noise schedules. The conditional transition probability is therefore $q(\boldsymbol{x}_t|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}_t;\alpha_t\boldsymbol{x},\sigma_t^2\boldsymbol{I})$. Taking the logarithm of this Gaussian density and computing its gradient with respect to $\boldsymbol{x}_t$ yields the analytical conditional score:

$$\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x}) = -\frac{\boldsymbol{x}_t-\alpha_t\boldsymbol{x}}{\sigma_t^2} \quad (31)$$

**Model Parameterization.** In practice, the neural network $\widehat{\boldsymbol{x}}$ is trained to directly predict the clean state $\boldsymbol{x}$ from $\boldsymbol{x}_t$. Following the standard parameterization in diffusion models, we express the score network $\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)$ in terms of this denoised prediction $\widehat{\boldsymbol{x}}$ as:

$$\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) = -\frac{\boldsymbol{x}_t-\alpha_t\widehat{\boldsymbol{x}}}{\sigma_t^2} \quad (32)$$

**Algebraic Simplification and Time Integration.** Substituting the analytical conditional score (Eq. 31) and the parameterized score network (Eq. 32) into the conditional score matching loss term, we obtain:

$$\begin{aligned}\|\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\|_2^2 &= \left\|-\frac{\boldsymbol{x}_t-\alpha_t\boldsymbol{x}}{\sigma_t^2} - \left(-\frac{\boldsymbol{x}_t-\alpha_t\widehat{\boldsymbol{x}}}{\sigma_t^2}\right)\right\|_2^2 \\ &= \left\|\frac{\alpha_t(\widehat{\boldsymbol{x}}-\boldsymbol{x})}{\sigma_t^2}\right\|_2^2 \\ &= \frac{\alpha_t^2}{\sigma_t^4}\|\widehat{\boldsymbol{x}}-\boldsymbol{x}\|_2^2 \end{aligned} \quad (33)$$

where the noisy vector $\boldsymbol{x}_t$ cancels out entirely. Integrating this simplified single-step objective over the continuous time interval $[0, T]$ yields:

$$\begin{aligned}\frac{1}{2}\int_0^T g_t^2 D_f(q(\boldsymbol{x}_t) \parallel p_\theta(\boldsymbol{x}_t))\mathrm{d}t &\equiv \frac{1}{2}\int_0^T g_t^2 \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\left[\|\nabla_{\boldsymbol{x}_t}\log q(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\|_2^2\right]\mathrm{d}t \\ &= \frac{1}{2}\int_0^T g_t^2 \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\left[\frac{\alpha_t^2}{\sigma_t^4}\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2\right]\mathrm{d}t \end{aligned} \tag{34}$$

By defining the time-dependent weighting parameter $w_t \triangleq \frac{g_t^2\alpha_t^2}{2\sigma_t^4}$, we arrive at the final equivalence:

$$\frac{1}{2}\int_0^T g_t^2 D_f\big(q(\boldsymbol{x}_t) \parallel p_\theta(\boldsymbol{x}_t)\big)\mathrm{d}t \equiv \int_0^T w_t \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{x}_t)}\big[\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2\big]\mathrm{d}t \tag{2}$$

This completes the mathematical proof of Eq. 2 in the paper.

### A.3 Proof of Discrepancy Contraction

In §3.1 of the paper, we assert that enforcing the source-level distribution alignment $p_\theta(\boldsymbol{x}) \approx q(\boldsymbol{x})$ establishes a uniform upper bound on the discrepancy of the marginalized distributions under the forward corruption process. Here, we present a mathematically rigorous proof of this contraction property. While the derivation is formulated using the Kullback-Leibler (KL) divergence, the same contraction holds generally for any $f$-divergence as the Data Processing Inequality (DPI) applies generally.

Let $K_t(\boldsymbol{x}_t|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}, (1-\bar{\alpha}_t)\boldsymbol{I})$ denote the transition kernel defined by the forward diffusion process. To ensure the relative entropy is well-defined, we assume absolute continuity of $q(\boldsymbol{x})$ with respect to $p_\theta(\boldsymbol{x})$ (denoted as $q \ll p_\theta$). Under this shared transition operator $K_t$, the marginalized probability densities at any intermediate timestep $t \in (0, T]$ for the true data distribution $q_0 = q(\boldsymbol{x})$ and the surrogate model distribution $p_0 = p_\theta(\boldsymbol{x})$ are given by the integral transforms:

$$q(\boldsymbol{x}_t) = \int q(\boldsymbol{x}_t|\boldsymbol{x})q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{35}$$

$$p_\theta(\boldsymbol{x}_t) = \int q(\boldsymbol{x}_t|\boldsymbol{x})p_\theta(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{36}$$

We construct the coupled joint distributions $(\boldsymbol{x}_t, \boldsymbol{x})$ for both processes as:

$$q(\boldsymbol{x}_t, \boldsymbol{x}) = q(\boldsymbol{x}_t|\boldsymbol{x})q(\boldsymbol{x}) \tag{37}$$

$$p_\theta(\boldsymbol{x}_t, \boldsymbol{x}) = q(\boldsymbol{x}_t|\boldsymbol{x})p_\theta(\boldsymbol{x}) \tag{38}$$

We emphasize that the same conditional transition kernel $q(\boldsymbol{x}_t|\boldsymbol{x})$ is employed under the model's joint distribution. This is because the forward corruption mechanism is purely physical and remains unaltered by the generative model; the model only specifies the initial surrogate distribution $p_\theta(\boldsymbol{x})$.

Since marginalization over $\boldsymbol{x}$ is a special case of applying a deterministic, measurable projection function $f(\boldsymbol{x}_t, \boldsymbol{x}) = \boldsymbol{x}_t$ to the joint random variables, the Data Processing Inequality (DPI) for KL divergence directly yields the contraction bound:

$$D_{\mathrm{kl}}(q(\boldsymbol{x}_t) \parallel p_\theta(\boldsymbol{x}_t)) \le D_{\mathrm{kl}}(q(\boldsymbol{x}_t, \boldsymbol{x}) \parallel p_\theta(\boldsymbol{x}_t, \boldsymbol{x})) \tag{39}$$

To evaluate the joint KL divergence on the right-hand side of Inequality equation 39 analytically, we expand the relative entropy definition:

$$D_{\mathrm{kl}}(q(\boldsymbol{x}_t, \boldsymbol{x}) \parallel p_\theta(\boldsymbol{x}_t, \boldsymbol{x})) = \int_{\mathcal{X}_0}\int_{\mathcal{X}_t} q(\boldsymbol{x}_t, \boldsymbol{x})\log\left(\frac{q(\boldsymbol{x}_t, \boldsymbol{x})}{p_\theta(\boldsymbol{x}_t, \boldsymbol{x})}\right)\mathrm{d}\boldsymbol{x}_t\mathrm{d}\boldsymbol{x}$$

Substituting the joint-to-conditional factorizations, we obtain:

$$D_{\mathrm{kl}}(q(\boldsymbol{x}_t, \boldsymbol{x}) \parallel p_\theta(\boldsymbol{x}_t, \boldsymbol{x})) = \int_{\mathcal{X}_0}\int_{\mathcal{X}_t} q(\boldsymbol{x}_t|\boldsymbol{x})q(\boldsymbol{x})\log\left(\frac{q(\boldsymbol{x}_t|\boldsymbol{x})q(\boldsymbol{x})}{q(\boldsymbol{x}_t|\boldsymbol{x})p_\theta(\boldsymbol{x})}\right)\mathrm{d}\boldsymbol{x}_t\mathrm{d}\boldsymbol{x} \tag{40}$$

Since the isotropic Gaussian transition kernel $q(\boldsymbol{x}_t|\boldsymbol{x})$ is strictly positive almost everywhere ($q(\boldsymbol{x}_t|\boldsymbol{x}) > 0$ a.e.) and shares the same support, the transition kernel terms cancel out within the logarithm. This algebraic cancellation yields:

$$\log\left(\frac{q(\boldsymbol{x}_t|\boldsymbol{x})q(\boldsymbol{x})}{q(\boldsymbol{x}_t|\boldsymbol{x})p_\theta(\boldsymbol{x})}\right) = \log\left(\frac{q(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right) \tag{41}$$

Applying Fubini's theorem (since the relative entropy is finite and well-defined), we factor the double integral as:

$$D_{\mathrm{kl}}(q(\boldsymbol{x}_t,\boldsymbol{x}) \parallel p_\theta(\boldsymbol{x}_t,\boldsymbol{x})) = \int_{\mathcal{X}_0}\left[\int_{\mathcal{X}_t} q(\boldsymbol{x}_t|\boldsymbol{x})\mathrm{d}\boldsymbol{x}_t\right] q(\boldsymbol{x})\log\left(\frac{q(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right)\mathrm{d}\boldsymbol{x} \tag{42}$$

By definition, the conditional transition density integrates to unity over its support, i.e., $\int_{\mathcal{X}_t} q(\boldsymbol{x}_t|\boldsymbol{x})\mathrm{d}\boldsymbol{x}_t = 1$ for all $\boldsymbol{x} \in \mathcal{X}_0$. Thus, the inner integral simplifies to:

$$D_{\mathrm{kl}}(q(\boldsymbol{x}_t,\boldsymbol{x}) \parallel p_\theta(\boldsymbol{x}_t,\boldsymbol{x})) = \int_{\mathcal{X}_0} q(\boldsymbol{x})\log\left(\frac{q(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right)\mathrm{d}\boldsymbol{x} = D_{\mathrm{kl}}(q(\boldsymbol{x}) \parallel p_\theta(\boldsymbol{x})) \tag{43}$$

Combining this result with Inequality 39, we establish the uniform contraction bound:

$$D_{\mathrm{kl}}(q(\boldsymbol{x}_t) \parallel p_\theta(\boldsymbol{x}_t)) \le D_{\mathrm{kl}}(q(\boldsymbol{x}) \parallel p_\theta(\boldsymbol{x})), \quad \forall\, t \in (0,T] \tag{44}$$

where the equality holds trivially at $t = 0$.

**Remark.** The above analysis treats $p_\theta(\boldsymbol{x}_t)$ as the push-forward measure of the model distribution $p_\theta(\boldsymbol{x})$ under the forward corruption kernel $q(\boldsymbol{x}_t|\boldsymbol{x})$. This corresponds to the distribution of noisy samples obtained by artificially corrupting the model's clean outputs. While this does not directly characterize the reverse-time sampling trajectory, it provides a rigorous worst-case guarantee for the latent distribution mismatch in the forward coupling process. Crucially, it demonstrates that minimizing the source-level discrepancy $D_{\mathrm{kl}}(q(\boldsymbol{x}) \parallel p_\theta(\boldsymbol{x}))$ via our MAL loss effectively controls the discrepancy at all intermediate noise levels *in the forward sense*, thereby preventing the propagated noisy predictions from drifting arbitrarily far from the true data manifold during the corruption process.

### A.4 Proof of Empirical Score Field via KDE

Suppose we have a mini-batch of $n_b$ samples $\{\boldsymbol{x}_i\}_{i=1}^{n_b}$ drawn from an unknown data distribution $q(\boldsymbol{x})$. Utilizing non-parametric Kernel Density Estimation (KDE), the empirical probability density $\widehat{q}(\boldsymbol{x})$ at any query point $\boldsymbol{x}$ can be approximated as:

$$\widehat{q}(\boldsymbol{x}) = \frac{1}{n_b}\sum_{i=1}^{n_b} k(\boldsymbol{x},\boldsymbol{x}_i) \tag{5}$$

where $k$ denotes the isotropic Gaussian kernel function:

$$k(\boldsymbol{x},\boldsymbol{x}_i) = \frac{1}{(2\pi\sigma^2)^{n_d/2}}\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma^2}\right) \tag{6}$$

with $n_d$ denoting the dimensionality of the feature space and $\sigma$ the kernel bandwidth.

**Chain Rule and Kernel Differentiation.** The score function is defined as the gradient of the log-density function with respect to the input $\boldsymbol{x}$. Applying the chain rule of calculus, we have:

$$\nabla_{\boldsymbol{x}}\log\widehat{q}(\boldsymbol{x}) = \frac{\nabla_{\boldsymbol{x}}\widehat{q}(\boldsymbol{x})}{\widehat{q}(\boldsymbol{x})} \tag{45}$$

To evaluate the numerator $\nabla_{\boldsymbol{x}}\widehat{q}(\boldsymbol{x})$, we compute the gradient of the summation:

$$\nabla_{\boldsymbol{x}}\widehat{q}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}}\left(\frac{1}{n_b}\sum_{i=1}^{n_b} k(\boldsymbol{x},\boldsymbol{x}_i)\right) = \frac{1}{n_b}\sum_{i=1}^{n_b}\nabla_{\boldsymbol{x}}k(\boldsymbol{x},\boldsymbol{x}_i) \tag{46}$$

Next, we calculate the gradient of an individual similarity weight $k(\boldsymbol{x}, \boldsymbol{x}_i)$ with respect to the query vector $\boldsymbol{x}$:

$$\nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}_i) = k(\boldsymbol{x}, \boldsymbol{x}_i) \cdot \nabla_{\boldsymbol{x}} \left( -\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma^2} \right) \tag{47}$$

By utilizing the standard vector derivative identity $\nabla_{\boldsymbol{x}} \|\boldsymbol{x} - \boldsymbol{a}\|^2 = 2(\boldsymbol{x} - \boldsymbol{a})$, the expression simplifies to:

$$\nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}_i) = k(\boldsymbol{x}, \boldsymbol{x}_i) \cdot \left( -\frac{2(\boldsymbol{x} - \boldsymbol{x}_i)}{2\sigma^2} \right) = \frac{k(\boldsymbol{x}, \boldsymbol{x}_i)}{\sigma^2} (\boldsymbol{x}_i - \boldsymbol{x}) \tag{48}$$

Substituting this kernel gradient back into Eq. 46, we obtain the analytical gradient of the empirical density function:

$$\nabla_{\boldsymbol{x}} \widehat{q}(\boldsymbol{x}) = \frac{1}{n_b} \sum_{i=1}^{n_b} \left[ \frac{k(\boldsymbol{x}, \boldsymbol{x}_i)}{\sigma^2} (\boldsymbol{x}_i - \boldsymbol{x}) \right] \tag{49}$$

**Closed-Form Score Representation.** Substituting the expressions of $\widehat{q}(\boldsymbol{x})$ and $\nabla_{\boldsymbol{x}} \widehat{q}(\boldsymbol{x})$ back into the score definition in Eq. 45 yields:

$$\nabla_{\boldsymbol{x}} \log \widehat{q}(\boldsymbol{x}) = \frac{\frac{1}{n_b \sigma^2} \sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{x})}{\frac{1}{n_b} \sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)} \tag{50}$$

By canceling the common constant factor $1/n_b$ and factoring out the scalar $1/\sigma^2$, we get:

$$\nabla_{\boldsymbol{x}} \log \widehat{q}(\boldsymbol{x}) = \frac{1}{\sigma^2} \left[ \frac{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{x})}{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)} \right] \tag{51}$$

Expanding the summation terms in the numerator allows us to separate the components:

$$\nabla_{\boldsymbol{x}} \log \widehat{q}(\boldsymbol{x}) = \frac{1}{\sigma^2} \left[ \frac{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)\boldsymbol{x}_i - (\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i))\, \boldsymbol{x}}{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)} \right] \tag{52}$$

By distributing the denominator, we arrive at the final closed-form representation of the empirical score function:

$$\nabla_{\boldsymbol{x}} \log \widehat{q}(\boldsymbol{x}) = \frac{1}{\sigma^2} \left( \frac{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)\boldsymbol{x}_i}{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)} - \boldsymbol{x} \right) = \frac{1}{\sigma^2} \left( \boldsymbol{\mu}(\boldsymbol{x}) - \boldsymbol{x} \right) \tag{53}$$

where $\boldsymbol{\mu}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)\boldsymbol{x}_i}{\sum_{i=1}^{n_b} k(\boldsymbol{x}, \boldsymbol{x}_i)}$ represents the weighted semantic centroid of the local neighborhood surrounding $\boldsymbol{x}$. Neglecting the scalar constant $1/\sigma^2$, this directly yields the score field formulation:

$$\nabla_{\boldsymbol{x}} \log \widehat{q}(\boldsymbol{x}) \propto \boldsymbol{\mu}(\boldsymbol{x}) - \boldsymbol{x} \tag{7}$$

# B Implementation Details

## B.1 Datasets and Evaluation Metrics

**Molecular Datasets (QM9 and ZINC250k).** We evaluate molecular generation on two standard benchmarks: QM9 (Ramakrishnan et al., 2014) and ZINC250k (Irwin et al., 2012). QM9 contains approximately 134k stable organic molecules with up to 9 heavy atoms from five elements (C, H, O, N, F), evaluating small-scale generation where electronic and chemical valency rules must be strictly respected. ZINC250k comprises around 250k larger, drug-like molecules with up to 38 heavy atoms extracted from the ZINC database, presenting a significantly more challenging setting characterized by higher structural diversity, longer chains, and polycyclic systems.

**General Graph Datasets (Planar, SBM, Enzymes, and Grid).** For general graph generation, we conduct experiments on four widely adopted benchmarks covering diverse topologies: Planar (Kipf & Welling, 2017) consists of synthetic planar graphs requiring strict non-crossing planar layouts; SBM (Holland et al., 1983) comprises community-structured graphs generated by the Stochastic

Block Model; Enzymes (Schomburg et al., 2004) contains real-world protein tertiary structures from bioinformatics; and Grid (You et al., 2018) consists of 2D regular grid graphs designed to evaluate the synthesis of rigid spatial regularities.

**Molecular Evaluation Metrics.** Following established protocols (Shi et al., 2020; Luo et al., 2021; Jo et al., 2022), we assess generated molecules without post-hoc correction using four metrics: chemical Validity (`Valid`), measuring the proportion of generated molecules passing RDKit sanitization; Fréchet ChemNet Distance (`FCD`) (Preuer et al., 2018), quantifying the Wasserstein-2 distance between hidden activations of ChemNet for real and generated molecules; Neighborhood Subgraph Pairwise Distance Kernel MMD (`NSPDK`) (Costa & De Grave, 2010), capturing graph-structural and label consistency; and Scaffold similarity (`Scaf.`), evaluating the distribution-level cosine similarity of extracted Bemis-Murcko scaffolds.

**General Graph Evaluation Metrics.** Following You et al. (2018) and Martinkus et al. (2022), we evaluate general graph generation by computing the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) over key graph statistics between generated and ground-truth test graphs. Specifically, we evaluate the degree distribution (`Deg.`), clustering coefficient distribution (`Clus.`), 4-node graphlet orbit counts (`Orbit`), and Laplacian spectrum eigenvalues (`Spec.`). Lower MMD values denote closer distributional proximity to the target manifold.

### B.2 Pretrained Graph Encoders

To project both real and generated graphs into a structured, semantically informed latent space, we employ domain-specific self-supervised graph encoders. By mapping graphs $G$ to latent representations $\boldsymbol{z} = \psi(G)$, we ensure that the kernel-based similarity computations in the KDE density estimation remain robust to high-frequency structural noise and focus on macro-structural semantics.

**Molecular Graphs (QM9 and ZINC250k).** To ensure the latent space captures meaningful structural priors, the encoder $\psi(\cdot)$
is pretrained using domain-specific self-supervised strategies. For molecular graphs, we employ the MolCLR (Wang et al., 2022) framework based on a GINE architecture, which jointly embeds atom attributes and bond semantics. During pretraining, each molecule is perturbed into two augmented views via stochastic node masking and edge deletion. For a mini-batch of $B$ molecules, the encoder is optimized by minimizing the NT-Xent loss over $2B$ augmented views:

$$\mathcal{L}_{\text{ntxent}} = \frac{1}{2B} \sum_{i=1}^{2B} -\log \frac{\exp\left(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq i]} \exp\left(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau\right)}, \tag{54}$$

where $j$ is the index of the positive counterpart for view $i$, $\text{sim}(\cdot)$ denotes cosine similarity, and $\tau$ is a temperature parameter. This consistency objective incentivizes the encoder to extract robust structural invariants by maximizing the agreement between perturbed views. Consequently, it induces a structured latent hypersphere where the angular proximity faithfully captures semantic and chemical resemblance, effectively mapping molecules with similar functional scaffolds to adjacent regions on the manifold.

**General Graphs (Planar, SBM, Enzymes, and Grid).** For general graphs, we adopt the GraphMAE (Hou et al., 2022) framework based on the GIN family. The pretraining involves a masked reconstruction task where 30% of node features are randomly masked. The encoder maps the partially observed graph into latent representations, from which a decoder attempts to reconstruct the original features. The entire model is optimized by minimizing the Scaled Cosine Error (SCE) on the masked nodes:

$$\mathcal{L}_{\text{sce}} = \sum_{i \in \mathcal{M}} \left(1 - \frac{\boldsymbol{x}_i^{\theta} \cdot \boldsymbol{x}_i}{\|\boldsymbol{x}_i^{\theta}\| \|\boldsymbol{x}_i\|}\right)^{\alpha}, \tag{55}$$

where $\mathcal{M}$ denotes the set of masked nodes, and $\alpha$ is a scaling exponent. This mechanism effectively re-weights the objective by suppressing loss contributions from well-reconstructed nodes while magnifying gradients for those with substantial errors. Consequently, the encoder is compelled to capture complex, high-order topological dependencies rather than overfitting to simple neighborhood redundancies.

### B.3 Backbone Integration and Plug-and-Play Setup

Our proposed Manifold Alignment (MAL) regularization is a plug-and-play framework designed to augment arbitrary graph diffusion backbones. Specifically, models such as GDSS and GruM perform generative modeling in the continuous graph representation space, whereas DiGress and DeFoG model graph structures directly in the discrete graph space. MAL is seamlessly integrated into these two paradigms as follows:

**Continuous-Space Models (GruM and GDSS).** We integrate MAL into continuous-space models based on their respective optimization paradigms. For GruM, the MAL loss $\mathcal{L}_{\text{mal}}$ is computed using the predicted clean states and is added directly as a regularizer to the standard MSE reconstruction loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mse}} + \lambda \mathcal{L}_{\text{mal}} \tag{56}$$

In GDSS, the baseline model is optimized using the Denoising Score Matching (DSM) objective on the continuous node-edge space. To compute the MAL loss, we reconstruct the estimated clean states $\widehat{X}_t$ and $\widehat{A}_t$ at timestep $t$ directly using Tweedie's formula. Specifically, given the continuous transition perturbation kernels $q(X_t \mid X) = \mathcal{N}(X_t; \alpha_{X,t}X, \sigma^2_{X,t}I)$ and $q(A_t \mid A) = \mathcal{N}(A_t; \alpha_{A,t}A, \sigma^2_{A,t}I)$, we can mathematically eliminate the injected Gaussian noise using the score functions to analytically solve for the clean targets:

$$\widehat{X}_t = \frac{X_t + \sigma^2_{X,t} s_{\theta,X}(X_t, A_t, t)}{\alpha_{X,t}} \tag{57}$$

$$\widehat{A}_t = \frac{A_t + \sigma^2_{A,t} s_{\theta,A}(X_t, A_t, t)}{\alpha_{A,t}} \tag{58}$$

where $\alpha_{X,t}$ and $\alpha_{A,t}$ denote the marginal mean coefficients induced by the integrated drift of the forward SDEs, while $\sigma^2_{X,t}$ and $\sigma^2_{A,t}$ represent the corresponding marginal transition variances determined by the diffusion processes. These reconstructed states are then used to compute the MAL loss $\mathcal{L}_{\text{mal}}$, which is added as a regularizer to the baseline DSM loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{dsm}} + \lambda \mathcal{L}_{\text{mal}} \tag{59}$$

**Discrete-Space Models (DiGress and DeFoG).** In these frameworks, the network directly models the categorical distributions of nodes and edges, optimized using the standard Cross-Entropy ($\mathcal{L}_{\text{ce}}$) loss. To compute the MAL loss while ensuring end-to-end differentiability for gradient backpropagation, we apply a softmax relaxation to convert the model's predicted categorical logits ($X_t^\theta$ and $A_t^\theta$) into continuous probability vectors:

$$\widehat{X}_t = \text{softmax}(X_t^\theta), \quad \widehat{A}_t = \text{softmax}(A_t^\theta) \tag{60}$$

The final optimized loss is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{mal}} \tag{61}$$

### B.4 Parameter Sensitivity

To systematically investigate the joint impact of key hyperparameters in MAL, we evaluate the generative performance under various combinations of the regularization weight $\lambda \in \{0.05, 0.1, 0.2\}$ and the density bandwidth parameter $\sigma \in \{0.1, 0.2, 0.3\}$ on the QM9 dataset, as shown in Figure 3. Across all three evaluated metrics, including `Valid`, `FCD`, and `Scaf.`, the combination of $\lambda = 0.10$ and $\sigma = 0.2$ achieves an optimal overall balance. Specifically, it yields the absolute lowest `FCD` value of 0.0970, which significantly outperforms the baseline score of 0.1240. Meanwhile, this setting boosts `Valid` to 0.9993 compared to the baseline of 0.9986, and `Scaf.` performance to 0.9372 compared to the baseline of 0.9357.

Across all settings, the density bandwidth $\sigma$ plays a decisive role in structural manifold alignment. When $\sigma$ is set to a small value such as $0.1$, the localized and sharp density fields lead to suboptimal performance, yielding an `FCD` of 0.1077 at $\lambda = 0.10$, mainly due to high-variance gradient signals during reverse-path alignment. Conversely, increasing $\sigma$ to $0.3$ leads to severe degradation, reaching

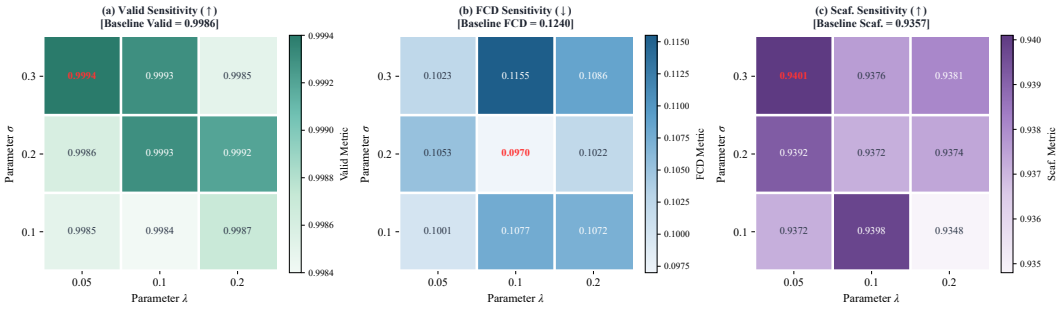



Figure 3: Sensitivity analysis of the regularization weight $\lambda$ and density bandwidth parameter $\sigma$ on the QM9 dataset across three evaluation metrics: (a) `Valid` (↑), (b) `FCD` (↓), and (c) `Scaf.` (↑). Red numbers highlight the optimal performance in each panel, with baseline scores provided at the top of each subfigure.

the worst `FCD` of 0.1155 at $\lambda = 0.10$. We attribute this performance drop to the over-smoothing of the estimated density field, which washes out fine-grained geometric details of the target manifold. Under the moderate bandwidth of $\sigma = 0.2$, the regularization weight $\lambda = 0.10$ exhibits a robust operating window, effectively guiding the reverse diffusion trajectory toward high-density regions of the real data distribution.

### B.5 Hyperparameter Settings

For the baseline diffusion models including GDSS, GruM, DiGress, and DeFoG, we strictly adhere to their original hyperparameters, optimization protocols such as Adam or AdamW optimizers, learning rate schedules, EMA decay, batch sizes, and training epochs to ensure a fair evaluation. The only exception is for GruM on the ZINC250k dataset, where we increased the training epochs from 500 to 1000 to achieve stable convergence.

| **Hyperparameter** | **QM9** | **ZINC250k** | **Planar** | **SBM** | **Enzymes** | **Grid** |
|---|---|---|---|---|---|---|
| Regularization weight ($\lambda$) | 0.1 | 0.2 | 0.1 | 0.3 | 0.2 | 0.2 |
| KDE kernel bandwidth ($\sigma$) | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Mini-batch size ($n_b$) | 2048 | 512 | 128 | 32 | 32 | 8 |

Table 5: Regularization and training hyperparameter settings for MAL.

The hyperparameter settings associated with MAL are categorized into two main groups. Table 5 summarizes the regularization and training-specific hyperparameters, including the regularization weight $\lambda$, the kernel density estimation bandwidth $\sigma$, and the mini-batch size $n_b$. The selection of these key parameters is grounded in principled design choices and dataset properties. Specifically, to ensure a fair evaluation and maintain stable optimization dynamics, the mini-batch size $n_b$ strictly retains the default batch size configurations of the baseline backbone models across all datasets. As demonstrated in the empirical sensitivity analysis in Figure 3, the KDE kernel bandwidth $\sigma$ is fixed to $0.2$ across all datasets because it strikes an optimal trade-off between avoiding high-variance gradients resulting from overly small bandwidths and preventing the over-smoothing of fine-grained manifold details induced by excessively large values. Meanwhile, the regularization weight $\lambda$ is tuned within a candidate search set $\lambda \in \{0.1, 0.2, 0.3\}$ based on the scale and structural complexity of the target dataset, where larger or more complex graph structures such as ZINC250k and SBM benefit from a slightly stronger manifold constraint with $\lambda$ set to 0.2 or higher.

Table 6 details the pretraining network configurations, specifying the encoder latent dimension $d$, the temperature $\tau$ for molecular contrastive pretraining, and the scaling exponent $\alpha$ for general graph pretraining. The choices for these pretraining configurations align with established representation learning standards and structural requirements. Specifically, the encoder latent dimension $d$ is set to $256$ for molecular graphs and expanded to $512$ for general graphs to accommodate their higher

| Hyperparameter | QM9 | ZINC250k | Planar | SBM | Enzymes | Grid |
|---|---|---|---|---|---|---|
| Encoder latent dimension ($d$) | 256 | 256 | 512 | 512 | 512 | 512 |
| Molecular pretraining temperature ($\tau$) | 0.1 | 0.1 | — | — | — | — |
| General pretraining scaling ($\alpha$) | — | — | 2.0 | 2.0 | 2.0 | 2.0 |

Table 6: Pretraining hyperparameter settings for MAL across different datasets.

topological complexity. For molecular contrastive pretraining, the temperature $\tau$ is set to $0.1$ following standard NT-Xent loss formulations to maintain appropriate gradient scaling for contrastive pairs. Lastly, the scaling exponent $\alpha$ is fixed to $2.0$ across general graph datasets to properly normalize structural graph features during pretraining.

| Metric | QM9 | ZINC250k | Planar | SBM | Enzymes | Grid |
|---|---|---|---|---|---|---|
| Training epochs | 1,000 | 1,000 | 30,000 | 30,000 | 5,000 | 5,000 |
| Training time (h) | $15.4 \pm 1.2$ | $50.2 \pm 2.5$ | $5.0 \pm 1.1$ | $28.3 \pm 2.8$ | $8.1 \pm 1.5$ | $11.6 \pm 2.3$ |
| Peak memory (GB) | $16.0 \pm 1.4$ | $54.2 \pm 2.1$ | $30.1 \pm 1.3$ | $54.0 \pm 2.6$ | $31.3 \pm 1.7$ | $59.2 \pm 2.4$ |

Table 7: Training overhead and resource consumption across datasets.

Table 7 details the computational overhead and resource requirements across all evaluated datasets, specifying the total training epochs alongside the wall-clock time and peak memory usage reported as mean $\pm$ std.

### B.6 Computing Environment and Reproducibility

To ensure complete reproducibility, all experiments in this work were conducted on a Linux server (x86_64) equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM), dual Intel(R) Xeon(R) Platinum 8268 CPUs @ 2.90GHz (48 physical cores and 96 logical threads in total), and 512 GB of DDR4 system memory. The underlying software environment is powered by Python 3.9.18, PyTorch 2.1.0+cu118, and CUDA 11.8 under NVIDIA driver version 550.90.07. Graph neural network architectures and molecular property evaluations are implemented using PyTorch Geometric (torch_geometric) version 2.6.1 and RDKit version 2023.03.3, respectively. All experimental results reported in Section 4, including both molecular and general graph generation benchmarks, were evaluated under a single execution run with a fixed random seed of 42 to guarantee deterministic execution and fair comparison across all models.

### B.7 Multi-Seed Performance Profile and Distribution Analysis

| Metric | Mean | Std | Median | Range [Min, Max] | 95% CI | IQR | Paired $t$-test ($p$) |
|---|---|---|---|---|---|---|---|
| Valid (↑) | 99.95% | $\pm$0.02% | 99.94% | [99.93%, 99.98%] | 99.95% $\pm$ 0.02% | 0.01% | 0.0016 |
| FCD (↓) | 0.1055 | $\pm$0.0048 | 0.1072 | [0.0971, 0.1087] | 0.1055 $\pm$ 0.0042 | 0.0022 | 0.0031 |
| NSPDK (↓) | 0.0002 | $\pm$0.0000 | 0.0002 | [0.0002, 0.0002] | 0.0002 $\pm$ 0.0000 | 0.0000 | 0.0001 |
| Scaf. (↑) | 0.9384 | $\pm$0.0013 | 0.9385 | [0.9365, 0.9401] | 0.9384 $\pm$ 0.0012 | 0.0010 | 0.2185 |

Table 8: Detailed statistical distribution profile of MAL across 5 random seeds on QM9, alongside paired $t$-test $p$-values compared to the GruM baseline.

To address the distribution variability and empirical stability across different random initializations, Table 8 provides a detailed statistical profile of our model on the QM9 dataset over 5 independent execution runs with random seeds $\{42, 43, 44, 45, 46\}$. In alignment with the primary evaluation metrics presented in Table 2 of the main text, we report the four core criteria: `Valid`, `FCD`, `NSPDK`, and `Scaf.`. Beyond standard central tendency statistics (Mean, Median), we explicitly report the Standard Deviation (Std), 95% Confidence Interval (CI), empirical Fluctuation Range [Min, Max], and Interquartile Range (IQR $= Q_3 - Q_1$). The extremely narrow fluctuation bounds and low variance across all metrics confirm that MAL consistently delivers stable geometric regularization without

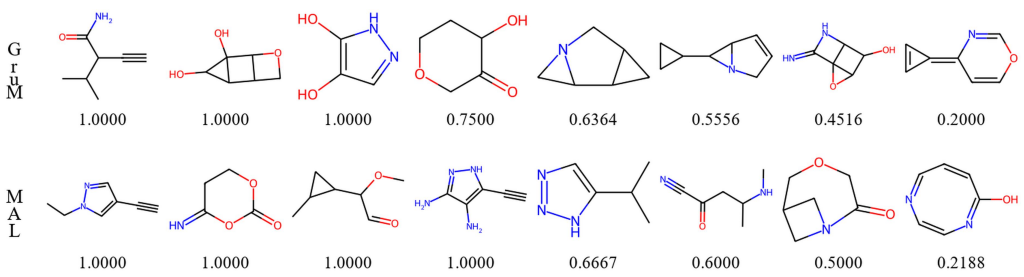



Figure 4: Qualitative comparison of generated molecules on QM9. The top row shows molecules generated by the baseline GruM, while the bottom row displays samples from our approach MAL. All molecules are sampled at identical index intervals from the second half of 10,000 generated samples to ensure an unbiased comparison. Annotations underneath each molecule represent Tanimoto similarity scores evaluated via Morgan fingerprints relative to their nearest training set neighbors.

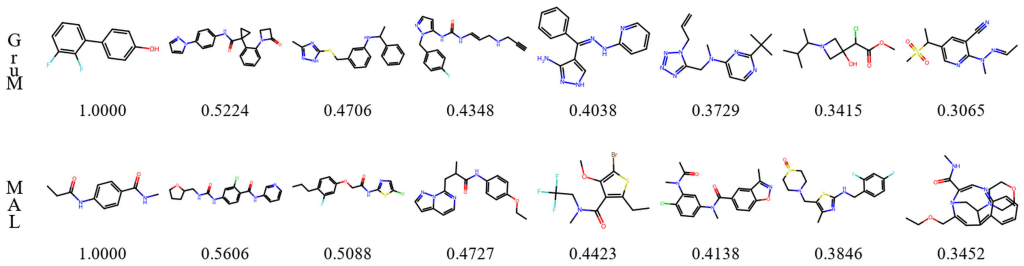



Figure 5: Qualitative comparison of generated molecules on ZINC250k. Top and bottom rows display samples from baseline GruM and our approach MAL, respectively. All molecules are deterministically sampled at equal index intervals of 1,250 from a 10,000-sample pool to eliminate selection bias. Annotations underneath denote Tanimoto similarity scores, with MAL consistently yielding higher scores and more cohesive, drug-like pharmacophores.

suffering from initialization-dependent variance. Remarkably, even under the worst-performing initialization (FCD of 0.1087 at seed 44), MAL still consistently outperforms the best reported baseline (DeFoG at 0.1102), further validating the deterministic robustness of our geometric alignment.

To assess whether the performance improvements of MAL over GruM are statistically significant rather than caused by random initialization variations, we perform paired hypothesis testing over five random seeds ($\{42, 43, 44, 45, 46\}$) on QM9. As reported in Table 8, MAL achieves statistically significant improvements over GruM on several key metrics. Specifically, the paired $t$-test yields $p = 0.0016$, $p = 0.0031$, and $p = 0.0001$ for `Valid`, `FCD`, and `NSPDK`, respectively, indicating that the observed improvements are unlikely to be attributed to random initialization alone. Meanwhile, `Scaf.` shows no statistically significant difference ($p = 0.2185$), demonstrating that MAL improves generation quality while preserving scaffold similarity.

# C Qualitative Visualization Results

## C.1 Visualization of Generated Molecular Graphs

To comprehensively demonstrate the superior topological fidelity and chemical validity of our proposed MAL framework, we perform qualitative visual evaluations across two standard molecular benchmark datasets: QM9 and ZINC250k. In both datasets, generated molecules are sampled deterministically without cherry-picking bias to illustrate how MAL preserves macro-structural semantics and prevents reverse-path trajectory drift into invalid off-manifold regions.

To provide a rigorous and unbiased qualitative evaluation of molecular generation quality on the QM9 dataset, Figure 4 presents a head-to-head visual comparison between the baseline GruM (Jo et al., 2024) in the top row and our proposed MAL framework in the bottom row. To strictly eliminate

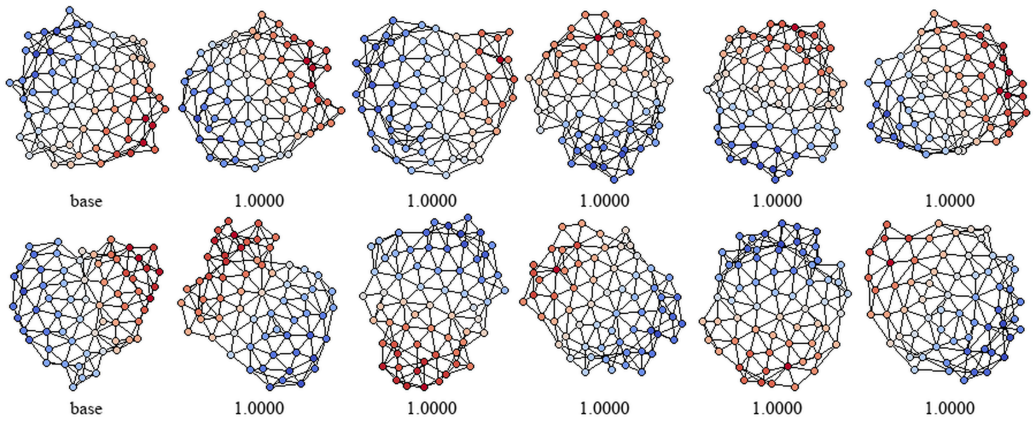



Figure 6: Nearest-neighbor similarity matching on Planar. The leftmost column displays representative generated base graphs produced by MAL. The subsequent five columns showcase the top-5 most structurally similar graphs retrieved from the training set, alongside their corresponding Tanimoto similarity scores annotated underneath each structure.

cherry-picking bias, all instances are deterministically sampled from the latter half, specifically the last 5,000 generated molecules, of a total generated pool of 10,000 samples for both models using identical index intervals across the similarity-ranked distribution. Annotated underneath each structure with its corresponding Tanimoto similarity score, the generated molecules span from a high-similarity match of 1.0000 down to lower similarity quantiles around 0.2000. Visual inspection reveals that MAL maintains high chemical validity and structural plausibility across diverse similarity regimes; for instance, in the second column, GruM generates a highly strained fused cyclobutane system, whereas MAL produces a stable six-membered heterocycle. Furthermore, in lower-similarity regions representing novel non-memorized topologies, such as the rightmost columns, while the baseline GruM produces strained rings or topologically awkward bond configurations, MAL preserves plausible functional groups and stable ring systems without severe unphysical strain, demonstrating that our batch-level manifold regularization successfully anchors reverse diffusion trajectories within the valid data manifold and prevents off-manifold drift.

To evaluate generative performance on complex, drug-like molecular structures, Figure 5 illustrates a qualitative comparison between the baseline GruM (Jo et al., 2024) in the top row and our proposed MAL framework in the bottom row on the ZINC250k dataset. To strictly eliminate selection bias, both rows display molecules deterministically sampled from a total pool of 10,000 generated samples at equal index intervals of 1,250 across the similarity-ranked distribution. Annotated underneath each structure with its corresponding Tanimoto similarity score, MAL consistently achieves higher similarity scores than GruM across all corresponding interval columns. Visual inspection demonstrates that MAL excels at synthesizing cohesive, highly drug-like scaffolds containing privileged medicinal chemistry motifs, such as robust amide linkers, halogenated aromatic rings, trifluoromethyl groups in column 5, and bioisosteric heterocycles including thiazoles and benzisoxazoles in columns 3, 6, and 7. In contrast, as molecular complexity increases, the baseline GruM frequently generates awkward aliphatic chains, unnatural hydrazine-like linkers, or fragmented side structures, demonstrating that our batch-level manifold regularization effectively guides the reverse diffusion process toward realistic, highly drug-like regions of the chemical manifold.

### C.2 Visualization of Generated General Graphs

To qualitatively demonstrate that MAL effectively pulls generated graphs back onto the valid data manifold while creating novel, non-memorized topologies, we perform nearest-neighbor similarity retrieval on general graph benchmarks, including the Planar and Stochastic Block Model (SBM) datasets.

Figure 6 illustrates representative base graphs generated by our MAL framework on Planar (leftmost column), alongside their top-5 most structurally similar graphs retrieved from the training

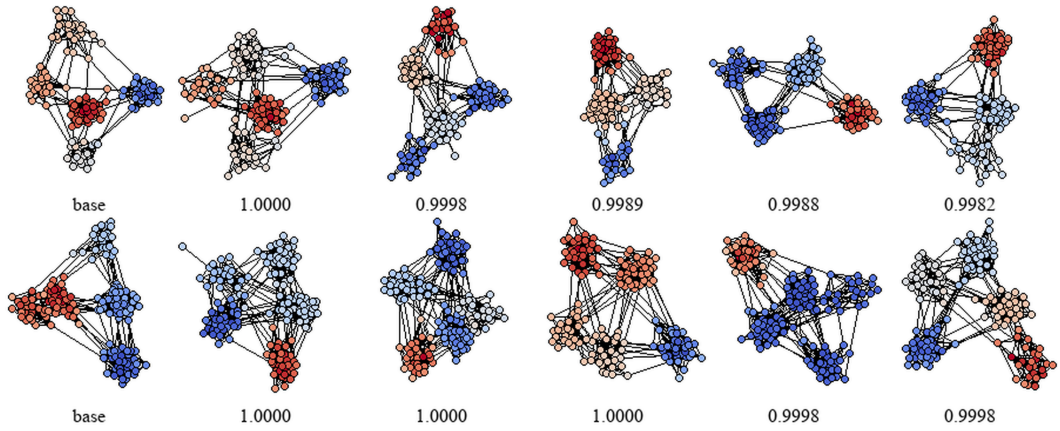



Figure 7: Nearest-neighbor similarity matching on SBM. The leftmost column displays generated base graphs produced by MAL, followed by the top-5 most structurally similar graphs retrieved from the training set with their Tanimoto similarity scores underneath. MAL accurately preserves intrinsic community cluster topologies and modular block structures.

set (subsequent five columns). Planar graphs impose strict topological constraints, requiring clean planar layouts without edge crossings. As depicted in Figure 6, the base graphs generated by MAL exhibit clean planar face divisions and well-distributed node layouts. Notably, while the retrieved training graphs achieve near-perfect similarity scores ($\approx 1.0000$) due to matching macro-topological properties (node/edge counts and planarity), a visual inspection confirms that the generated graphs remain structurally non-identical to their nearest training neighbors. This demonstrates that MAL successfully grounds the generative trajectory strictly within the valid planar data manifold, while effectively synthesizing novel and non-memorized graph instances rather than trivially replicating training samples.

Figure 7 shows the similarity retrieval results on SBM. Its topologies are defined by modular community clusters characterized by dense intra-cluster connections and sparse inter-cluster links. As illustrated in Figure 7, the base graphs generated by MAL accurately preserve distinct community structures and clear cluster boundaries. The top-5 retrieved training graphs yield consistently high similarity scores (ranging from $0.9982$ to $1.0000$). These high similarity values verify that MAL successfully eliminates topological noise and enforces global block-structure consistency across generated instances.