Graph Energy Matching: Transport-Aligned Energy-Based Modeling for Graph Generation

Michal Balcerak[1,2,†] michal_balcerak@harvard.edu **Suprosanna Shit**[3] **Chinmay Prabhakar**[3]

Sebastian Kaltenbach[1] **Michael S. Albergo**[1,2] **Yilun Du**[1,2,*] **Bjoern Menze**[3,*]

[1]Harvard University [2]Kempner Institute
[3]University of Zurich

Abstract

Generative modeling of discrete data, such as graphs, underpins many scientific and industrial applications, including molecular discovery and materials design. In these domains, probabilistic inference is particularly valuable, as it enables composable generation and principled incorporation of desired constraints, such as structural or functional properties. Energy-based models naturally support this goal by capturing relative likelihoods and enabling composable inference by directly enforcing constraints during inference. However, discrete energy-based models typically struggle with efficient and high-quality sampling, as off-support regions often contain spurious local minima, trapping samplers and causing training instabilities, resulting in a fidelity gap compared to discrete diffusion models. To address this gap, we introduce *Graph Energy Matching (GEM)*, a discrete generative framework inspired by the Jordan–Kinderlehrer–Otto (JKO) transport-map optimization perspective. GEM learns a permutation-invariant potential energy that simultaneously guides discrete transport from noise toward high-likelihood graph regions and refines samples within these regions. We further introduce a sampling protocol leveraging an energy-based switching strategy, seamlessly bridging rapid, gradient-guided transport and a local mixing regime for effective exploration. On molecular graph benchmarks, GEM matches or surpasses strong discrete diffusion baselines on most reported metrics. Beyond improving generation quality, GEM's relative likelihood modeling enables targeted exploration, facilitating compositional generation, property-constrained sampling, and interpolation between graphs. Project page: https://michalbalcerak.ai/graph-energy-matching/.

1 Introduction

Generative modeling of discrete objects, such as graphs, is a central challenge across many scientific and industrial applications, including drug discovery and materials design [Jin et al., 2018, Stokes et al., 2020, Xie et al., 2022]. While domain-driven simulations explicitly encode physical knowledge [Noé et al., 2020], they quickly become intractable due to the combinatorial complexity of the underlying data distribution [Polishchuk et al., 2013], highlighting the need for generative models that efficiently capture implicit, domain-specific patterns.

[†]Much of this work was done during the author's previous appointment at the University of Zurich.
[*]Contributed equally as senior authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Current state-of-the-art models for unconditional graph generation predominantly rely on discrete diffusion-like processes, implemented either as discrete-time denoising steps [Vignac et al., 2023] or continuous-time Markov chains (CTMCs) [Campbell et al., 2024, Siraudin et al., 2025, Qin et al., 2025]. These methods generate graphs by progressively denoising samples from a simple prior distribution toward the target data distribution. For instance, DeFoG [Qin et al., 2025] learns CTMC transition rates to remove noise from corrupted graph states, while Variational Flow Matching (VFM) [Eijkelboom et al., 2024, 2025] parameterizes stochastic trajectories that continuously transform samples from noise distributions into clean graphs. Although these approaches yield high-quality unconditional samples, they define distributions implicitly via noisy intermediate states, lacking an explicit model directly on clean graphs. This implicit representation complicates the enforcement of user-specified properties or constraints, as guidance must operate on intermediate, often off-manifold states, where predictors and conditioning signals become less reliable [Vignac et al., 2023].

Energy-based models (EBMs) [Hinton, 2002, LeCun et al., 2007, Du and Mordatch, 2019] offer a complementary perspective that naturally addresses these limitations. EBMs represent relative probability structure through a scalar energy function, enabling direct incorporation of constraints, priors, or property-based objectives at inference time without retraining [Du, 2025]. For graph generation, EBMs offer particular advantages, as graph-structured constraints and domain-specific priors can be naturally integrated into the energy function. Historically, however, EBMs have struggled with sample quality, primarily due to poor sampling efficiency in discrete, combinatorial domains, as exemplified by existing methods such as GraphEBM [Liu et al., 2021]. Consequently, discrete EBMs have predominantly served as scoring models for out-of-distribution detection rather than competitive generation [Wu et al., 2023, Fuchsgruber et al., 2024].

Energy Matching [Balcerak et al., 2025] bridges these perspectives by building on the Jordan–Kinderlehrer–Otto (JKO) formulation [Jordan et al., 1998] and its first-order optimality perspective [Terpin et al., 2024, Lanzetti et al., 2024, 2025], coupling transport toward the data distribution with refinement through a time-indexed sequence of sampling steps. In these continuous formulations, dynamics typically follow ordinary differential equations (ODEs) or stochastic differential equations (SDEs), guided by a learned scalar potential that defines both the transport direction and refinement by capturing a Boltzmann-like probability structure of the data.

However, Energy Matching formulations operate in continuous spaces and rely on time-indexed dynamics defined over ambient continuous domains. This approach does not naturally extend to discrete spaces, where generation fundamentally requires discrete sampling characterized by abrupt, local state transitions rather than continuous trajectories. This highlights the need for a new framework enabling high-quality graph generation while explicitly capturing the relative likelihood structure, thus recovering the compositional and constraint-handling advantages of EBMs.

Contributions. We introduce *Graph Energy Matching (GEM)*, a novel discrete energy-based generative model for graphs, achieving molecular graph generation quality that matches or surpasses leading discrete diffusion approaches on most reported metrics. Beyond unconditional sampling, the learned relative-likelihood structure enables GEM to incorporate compositional constraints at inference time and facilitates the computation of geodesics between graphs.

We leverage a JKO-style transport map optimization perspective to define (i) a transport-aligned discrete proposal to rapidly move towards high-probability graphs, and (ii) a discrete mixing proposal that efficiently explores the learned data geometry through local graph edits. GEM provides a principled pathway toward solving limitations of energy-based discrete graph generation, paving the way for their broader applicability.

2 Preliminaries: Energy Matching and JKO

Energy matching learns a scalar potential $V_\theta : \mathbb{X} \to \mathbb{R}$, parameterized by θ, on a continuous data space $\mathbb{X}$. It simultaneously serves two complementary roles: (i) off-manifold guidance: directing samples toward the data manifold via the gradient field $-\nabla_x V_\theta$, and (ii) near-equilibrium refinement: characterizing the data distribution as an EBM.

These objectives decompose model learning into a *global* transport-alignment term and a *local* density-refinement term. A time-dependent temperature schedule (e.g., $\epsilon(t) = 0$ for $0 \leq t < 1$ for

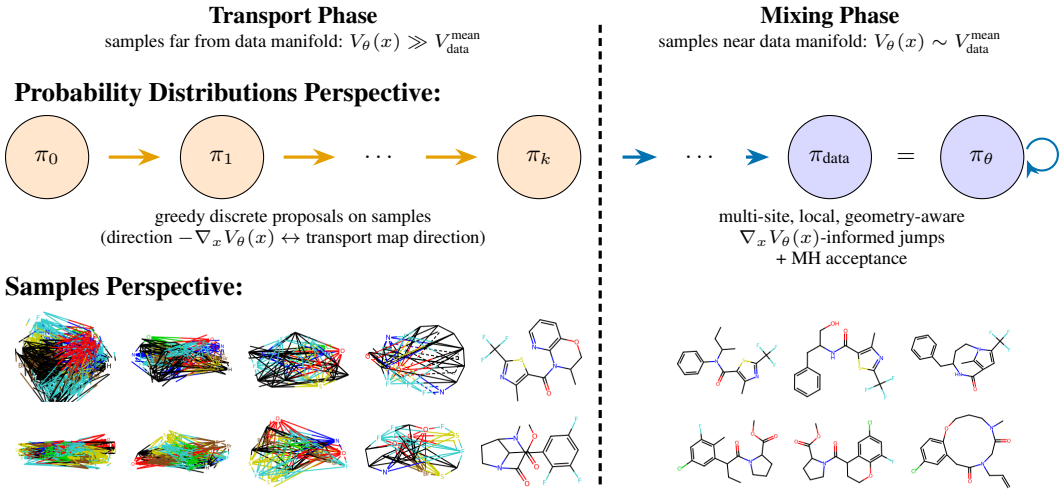

Figure 1: **GEM Sampling Overview.** Two perspectives on GEM sampling: a probability-distribution view (top) of the two-phase MCMC process, and a samples view (bottom) showing molecular trajectories from MOSES. Sampling alternates between a transport phase, where gradient-informed, greedy proposals rapidly move samples toward regions of high probability, and a mixing phase employing MH acceptance to ensure correct stationary distribution and efficient mixing between modes. **Color key:** transport (orange), MH (blue).

transport and $\epsilon(t) = \epsilon$ for $t \geq 1$ for mixing, promoting exploration) ensures that samples converge to the Gibbs measure $\rho_\theta(x) \propto \exp(-\frac{1}{\epsilon} V_\theta(x))$ as $t \to \infty$.

Energy Matching can be viewed through an optimization-in-probability-spaces perspective, expressed via the JKO scheme from density ρ_t to $\rho_{t+\Delta t}$:

$$\rho_{t+\Delta t} = \arg\min_{\rho} \frac{1}{2\Delta t} \inf_{\gamma \in \Gamma(\rho_t, \rho)} \mathbb{E}_{(x,y)\sim\gamma}[c(x,y)] + \int V_\theta(y)\, \rho(y)\, \mathrm{d}y + \epsilon(t) \int \rho(y) \log \rho(y)\, \mathrm{d}y, \tag{2.1}$$

where the cost $c(x, y)$ quantifies displacement from x to y, typically set as $c(x, y) = \|x - y\|^2$ in continuous spaces.

The coupling $\gamma \in \Gamma(\rho_t, \rho)$ is a joint distribution that characterizes a transport plan. Intuitively, $\gamma(x, y)$ encodes "how much mass from each location $x \sim \rho_t$ is transported to each new location $y \sim \rho$", thus determining the optimal reassignment of mass that minimizes the expected transportation cost $\mathbb{E}_{(x,y)\sim\gamma}[\, c(x, y)\,]$. In practice, couplings γ can be precomputed or estimated during training, and V_θ is optimized accordingly, aligning it with the induced transport trajectories.

Sampling (two regimes). Sampling starts from a transport-aligned drift and switches to refinement via a prescribed time-dependent temperature $\epsilon(t)$. In continuous time, this corresponds to the time-inhomogeneous SDE:

$$\mathrm{d}x_t = -\nabla_{x_t} V_\theta(x_t)\, \mathrm{d}t + \sqrt{2\,\epsilon(t)}\, \mathrm{d}W_t, \tag{2.2}$$

which reduces to an ODE when $\epsilon(t) = 0$ and becomes stochastic as $\epsilon(t) > 0$. In practice, sampling corresponds to explicitly integrating (2.2) over a fixed time interval.

3 Graph Energy Matching

We extend the continuous Energy Matching formulation to discrete graph spaces by introducing two novel discrete proposals: one enabling transport-aligned rapid movements across graph spaces, and the other facilitating mixing within high-probability regions. Our formulation employs an energy-based switching mechanism that clearly separates *transport* and *mixing* sampling regimes without requiring explicit indexing of time or noise levels. Sampling is efficiently facilitated through

deterministic greedy proposals in the transport regime and stochastic gradient-informed proposals during mixing. The sampler uses a temperature ϵ to control stochasticity: the transport regime is the deterministic $\epsilon \to 0$ limit, while the mixing regime uses $\epsilon > 0$. Further, we propose a temperature-annealing strategy to efficiently generate novel samples from given data samples.

Graph representation. We consider a discrete state space $\mathcal{X} := \left(\mathcal{X}_{\text{node}}\right)^n \times \left(\mathcal{X}_{\text{edge}}\right)^{\binom{n}{2}}$ to represent undirected graphs. A graph $x \in \mathcal{X}$ contains a set of n nodes $\mathcal{V} = \{1, \ldots, n\}$ and edges $\mathcal{E}$ defining the connectivity between these nodes. Each node $i \in \mathcal{V}$ has a node class $x_i \in \mathcal{X}_{\text{node}}$ (e.g. an atom type), and each edge $(i, j) \in \mathcal{E}$ has an edge class $x_{ij} \in \mathcal{X}_{\text{edge}}$ (e.g. a bond type). The edge space $\mathcal{X}_{\text{edge}}$ includes an element indicating the absence of an edge. The graph $x := \left(x_{\mathcal{V}}, x_{\mathcal{E}}\right)$ consists of $x_{\mathcal{V}}$ and $x_{\mathcal{E}}$ denoting categorical node and edge features, respectively, and are defined as:

$$x_{\mathcal{V}} := [x_i]_{i=1}^n \in (\mathcal{X}_{\text{node}})^n; \quad x_{\mathcal{E}} := [x_{ij}]_{i<j} \in \left(\mathcal{X}_{\text{edge}}\right)^{\binom{n}{2}} \tag{3.1}$$

We embed the node and edge categorical feature vectors $x_{\mathcal{V}}$ and $x_{\mathcal{E}}$ using a real-valued one-hot encoding and treat them as continuous features via:

$$\phi_{\mathcal{V}} : \mathcal{X}_{\text{node}} \to \mathbb{R}^{l_{\text{node}}}; \quad \phi_{\mathcal{E}} : \mathcal{X}_{\text{edge}} \to \mathbb{R}^{l_{\text{edge}}}$$

where $l_{\text{node}} = |\mathcal{X}_{\text{node}}|$ and $l_{\text{edge}} = |\mathcal{X}_{\text{edge}}|$ denote the numbers of node and edge classes, respectively. We denote the real-valued node features $\hat{x}_{\mathcal{V}} \in \mathbb{R}^{nl_{\text{node}}}$ and edge features $\hat{x}_{\mathcal{E}} \in \mathbb{R}^{\binom{n}{2} l_{\text{edge}}}$, and the corresponding graph representation $\hat{x} \in \mathbb{R}^d$, where $d = nl_{\text{node}} + \binom{n}{2} l_{\text{edge}}$, as:

$$\hat{x}_{\mathcal{V}} := \oplus_{i=1}^n \phi_{\mathcal{V}}(x_i); \quad \hat{x}_{\mathcal{E}} := \oplus_{i<j} \phi_{\mathcal{E}}(x_{ij}); \quad \hat{x} := \hat{x}_{\mathcal{V}} \oplus \hat{x}_{\mathcal{E}}. \tag{3.2}$$

where $\oplus$ denotes the column-wise concatenation operation.

Local and Permutation-Invariant Cost. We measure local discrepancies between two graphs x and y using the embedding-space cost $c(x, y) := \tilde{c}(\hat{x}, \hat{y})$, which is differentiable in both arguments. The potential V_θ is permutation-invariant by construction (Appendix A). A permutation $\sigma \in S_n$ acts on node indices and induces a relabeling of the graph: $(\sigma \cdot x)_i = x_{\sigma(i)}$ and $(\sigma \cdot x)_{ij} = x_{\sigma(i)\sigma(j)}$. To obtain a permutation-invariant matching, we first define a local cost (c_{loc}) as:

$$c_{\text{loc}}(x, y) \;=\; \lambda_{\mathcal{V}} \|\hat{x}_{\mathcal{V}} - \hat{y}_{\mathcal{V}}\|_2^2 + \lambda_{\mathcal{E}} \|\hat{x}_{\mathcal{E}} - \hat{y}_{\mathcal{E}}\|_2^2, \tag{3.3}$$

with $\lambda_{\mathcal{V}}, \lambda_{\mathcal{E}} > 0$ scaling node and edge contributions. Optimizing over node relabelings lifts the local cost to the hard-permutation Fused Gromov–Wasserstein (FGW) cost [Vayer et al., 2019]:

$$c_{\text{FGW}}(x, y) = \min_{\sigma \in S_n} \{c_{\text{loc}}(x, \sigma \cdot y)\}. \tag{3.4}$$

In practice, we predominantly use the local cost c_{loc}, as it provides a valid and efficient measure for scoring pairs of neighboring graphs, e.g. when evaluating displacements induced by local jump proposals. When permutation-invariant comparisons between non-local graph pairs are required, we instead employ the c_{FGW} cost. Exact computation is generally intractable, as it is linked to the Quadratic Assignment Problem (QAP) [Vayer et al., 2019], so we use fast approximations: histogram matching for noisy–clean pairs and node matching for molecule–molecule pairs (Appendix C).

Learnable Components. The primary learnable component of our framework is the scalar energy potential $V_\theta(x)$, implemented via a neural network. While we evaluate V_θ exclusively on discrete graphs $x \in \mathcal{X}$, we leverage the continuous embedding representation $\hat{x}$ to enable gradient-based computations. Specifically, the differentiability of $V_\theta(x) := \tilde{V}_\theta(\hat{x})$ with respect to $\hat{x}$ allows us to compute gradients that directly inform and guide local jump proposals used by the sampler (see Sections 3.1 and 3.2). Thus, the continuous embedding bridges the discrete graph domain with gradient-based optimization.

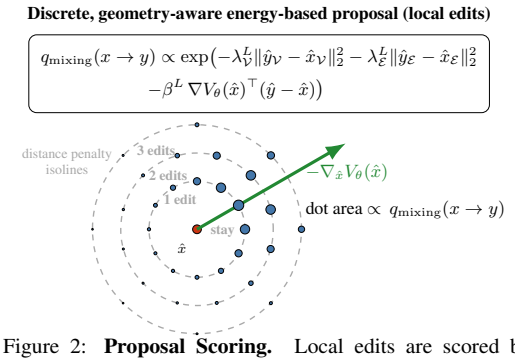

Figure 2: **Proposal Scoring.** Local edits are scored by $q_{\text{mixing}}(x \to y)$; dot area encodes proposal probability. Gradient-aligned, shorter jumps are favored, and stay proposals are resampled.

3.1 Transport-Aligned Proposal ($\epsilon \to 0$)

Let $(\pi_k)_{k=0}^K$ denote intermediate discrete distributions along a path from an initial noise distribution to the data distribution. Given an optimal coupling γ_k (approximated in practice by tractable methods, e.g., Albergo et al. [2024], Tong et al. [2024]) and its induced transport map T_k, we assume a consistent embedding-based representation $y = T_k(x) \iff \hat{y} = \tilde{T}_k(\hat{x})$, with $x \sim \pi_k$ and $y = T_k(x) \sim \pi_{k+1}$ denoting consecutive graph states along the transport, and $\tilde{T}_k$ a continuous map realizing T_k in the embedded space. Expressing the JKO variational objective from (2.1) as an optimization over transport maps (see Appendix B for derivation) and substituting the time-step parameter Δt with the jump step-size η yields:

$$T_k(x) \in \arg\min_{y \in \mathcal{X}} \left\{ \frac{1}{2\eta} c(x, y) + V_\theta(y) \right\}, \tag{3.5}$$

where $\pi_{k+1} = T_k^{\#} \pi_k$ denotes the pushforward of π_k. Assuming the minimizer $T_k(x)$ (3.5) lies in the interior, the first-order optimality condition becomes:

$$0 = \nabla_{\hat{y}} \left[\frac{1}{2\eta} \tilde{c}(\hat{x}, \hat{y}) + \tilde{V}_\theta(\hat{y}) \right] \Big|_{\hat{y} = \tilde{T}_k(\hat{x})} \implies \nabla_{\hat{y}} \tilde{V}_\theta\Big(\tilde{T}_k(\hat{x})\Big) = -\frac{1}{2\eta} \nabla_{\hat{y}} \tilde{c}\Big(\hat{x}, \tilde{T}_k(\hat{x})\Big). \tag{3.6}$$

This condition characterizes the solution and defines the training objective (Section 3.4). During sampling, given a trained potential $\tilde{V}_\theta$ that satisfies the optimality objective (3.6), our goal is to approximately solve (3.5) using a local proposal mechanism. Instead of performing large global moves, we iteratively descend the potential by making local edits, enabling a Taylor expansion of $\tilde{V}_\theta(\hat{y})$ around the current state $\hat{x}$. Rather than controlling the step-size parameter η, we simplify the (local) greedy proposal by restricting the kernel to exactly N edits per jump. Since our jumps are local, we use the cost $c_{\text{loc}}(\hat{x}, \hat{y})$ with distinct weights $\lambda_{\mathcal{V}}$ and $\lambda_{\mathcal{E}}$ (3.3).

Linearizing around $\hat{x}$: we approximate $\tilde{V}_\theta(\hat{y}) \approx \tilde{V}_\theta(\hat{x}) + (\hat{y} - \hat{x})^\top \nabla_{\hat{x}} \tilde{V}_\theta(\hat{x})$. We deterministically select the candidate y from the candidate set $C_N(x)$, defined as the set of graphs reachable from x via exactly N *discrete* modifications (edge changes/removals or node type alterations), by minimizing the sum of the potential change and the transport cost:

$$y^*(x) = \arg\min_{y \in C_N(x)} \left((\hat{y} - \hat{x})^\top \nabla_{\hat{x}} V_\theta(\hat{x}) + c_{\text{loc}}(x, y) \right), \quad q_{\text{greedy}}(x \to y) = \mathbb{I}_{\{y = y^*(x)\}} \tag{3.7}$$

In the transport regime, proposed moves are deterministically projected onto directions associated with strictly negative gradients, aiming for a decrease in energy. Edits not meeting this criterion are rejected. If no candidate yields a negative gradient, the greedy sampler stalls, switching to the Langevin-based proposal q_{mixing} in Section 3.2. More about proposal schedules in Section 3.3. For further proposal implementation details, see Appendix D.

3.2 Discrete Mixing Proposal ($\epsilon > 0$)

Discretizing dynamics (2.2) in embedding space with temperature ϵ, step η, and graphs $x, y \in \mathcal{X}$:

$$q_{\text{mixing}}(x \to y) \propto \exp\left(-\frac{1}{4\epsilon\eta} \|\hat{y} - \hat{x} + \eta \nabla_{\hat{x}} V_\theta(\hat{x})\|^2 \right). \tag{3.8}$$

We expand the mixing transition probability explicitly in terms of $\hat{y}$, simplifying by removing terms independent of $\hat{y}$ and substituting the squared-distance term $\|\hat{y} - \hat{x}\|^2$ with the local graph distance $\tilde{c}_{\text{loc}}(\hat{x}, \hat{y})$. This gives:

$$\log q_{\text{mixing}}(x \to y) = -\frac{1}{4\epsilon\eta} \tilde{c}_{\text{loc}}(\hat{x}, \hat{y}) - \frac{1}{2\epsilon} \nabla_{\hat{x}} V_\theta(\hat{x})^\top (\hat{y} - \hat{x}) + \text{const}. \tag{3.9}$$

The proposal is effectively local due to the distance penalty, though, in principle, it can still propose an arbitrary number of edits, enabling mode escape. We define Langevin-specific parameters $(\lambda_{\mathcal{V}}^L, \lambda_{\mathcal{E}}^L, \beta^L)$ by absorbing all constants, including ϵ, η, and the scaling terms $\lambda_{\mathcal{V}}, \lambda_{\mathcal{E}}$ from c_{loc}:

$$q_{\text{mixing}}(x \to y) \propto \exp\big(-\lambda_{\mathcal{V}}^L \|\hat{y}_{\mathcal{V}} - \hat{x}_{\mathcal{V}}\|_2^2 - \lambda_{\mathcal{E}}^L \|\hat{y}_{\mathcal{E}} - \hat{x}_{\mathcal{E}}\|_2^2 - \beta^L \nabla_{\hat{x}} V_\theta(\hat{x})^\top (\hat{y} - \hat{x}) \big). \tag{3.10}$$

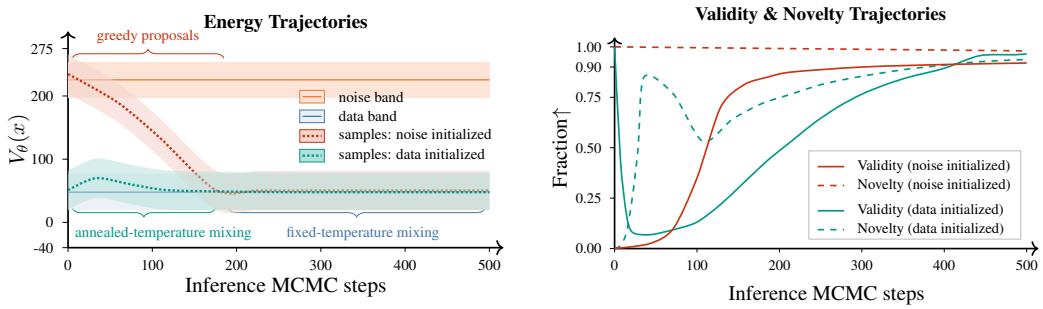

Figure 3: **Energy and Sampling Trajectories.** Left: energy evolution for noise- vs. data-initialized chains, with mean $\pm$ 1 std and reference energies $\mathbb{E}_{x\sim\pi_{\text{data}}}[V_\theta(x)]$ and $\mathbb{E}_{x\sim\pi_0}[V_\theta(x)]$. Noise-initialized chains use greedy proposals to reach the data distribution, with these transport-aligned steps serving as an effective burn-in for the subsequent mixing chain; data-initialized chains use temperature annealing (low initial β_{mh}, then increasing) to recover novelty. Right: validity and novelty trajectories for data-initialized (teal) and noise-initialized (red) chains. Uniqueness $\approx$100%.

The resulting kernel can be viewed as a *linearized locally balanced proposal* [Zanella, 2020]; related discrete-gradient samplers appear in [Grathwohl et al., 2021, Zhang et al., 2022] for particular costs or candidate sets. Figure 2 illustrates the resulting geometry-aware proposal scoring and how gradient alignment competes with distance penalties.

In practice, we pair the q_{mixing} proposal with a Metropolis–Hastings (MH) acceptance step to ensure convergence to the target distribution $\pi_\theta(x) \propto \exp(-\beta_{\text{mh}} V_\theta(x))$. In the mixing regime, the stochastic proposal is accepted with the following MH probability:

$$\alpha(x, y) = \min\left(1, \frac{\pi_\theta(y)\, q_{\text{mixing}}(y \to x)}{\pi_\theta(x)\, q_{\text{mixing}}(x \to y)}\right). \tag{3.11}$$

This choice enforces detailed balance:

$$\pi_\theta(x)\, q_{\text{mixing}}(x \to y)\, \alpha(x, y) = \pi_\theta(y)\, q_{\text{mixing}}(y \to x)\, \alpha(y, x) \tag{3.12}$$

so π_θ is stationary for the mixing dynamics [Metropolis et al., 1953, Hastings, 1970]. Implementation details of the proposal kernel are provided in Appendix D. We calibrate the parameters β_{mh}, β^L, $\lambda_{\mathcal{V}}^L$, and $\lambda_{\mathcal{E}}^L$ by maximizing the relative likelihood of generated samples under the learned energy V_θ, directly after the warmup training phase described in Section 3.4.

3.3 Initializations and Proposal Schedules

Determining the Regime. We control the regime through an energy-based binary indicator $s \in \{0, 1\}$: $s = 0$ selects transport (deterministic $\epsilon \to 0$ limit), while $s = 1$ selects mixing (finite $\epsilon > 0$). If initialized at noise, the sampler follows deterministic greedy edits guided strictly by negative gradient directions. Transition from transport to mixing occurs when the Markov chain either reaches the target energy (defined as the mean energy of the training samples estimated during training) or becomes trapped in a local minimum (i.e., no valid gradient-improving edits exist for all samples), whichever happens first. Once triggered, the sampler enters the stochastic mixing regime.

Sample Initialization. We use two initialization regions with aligned proposal schedules:

- **Noise initialization:** sample the node count from the empirical histogram, then draw node/edge types uniformly (uniform noise). These start far from the data distribution, so we run transport with greedy proposals q_{greedy} until the energy-based switch, then enter mixing with q_{mixing}.
- **Data initialization:** draw graphs from the training set. We skip the transport phase and start in the mixing regime with q_{mixing}, initially using a higher model temperature (lower β_{mh}) to increase the acceptance rate, and then annealing (raising β_{mh}) before fixed-temperature mixing (Appendix D).

Figure 3 summarizes the resulting energy trajectories under noise vs. data initialization. Further energy diagnostics are provided in Appendix F.5.

3.4 Training objectives

Noise and Data Interpolation via Minibatch coupling. Given minibatches $x^0 \sim \pi_0$ and $x^{\text{data}} \sim \pi_{\text{data}}$, where π_0 is the uniform noise and π_{data} is the data distribution respectively, we pair samples using a permutation-invariant approximation of $c_{\text{FGW}}(x^0, x^{\text{data}})$ described in Section 3. Specifically, we use histogram matching (Appendix C) and solve for a map T induced by the minibatch coupling $\tilde{\gamma}$ using the `POT` solver [Flamary et al., 2021]. Source graphs x^0 are paired with data graphs $x^{\text{data}} = T(x^0)$.

Crucially, we utilize c_{FGW} solely to select optimal pairs based on global structural similarity (e.g., node and edge counts). Denote their embeddings by $\hat{x}^0$ and $\hat{x}^{\text{data}}$. Define the displacement in the embedding space as $v = \hat{x}^{\text{data}} - \hat{x}^0$. We sample a *discrete* interpolant $x^t := (x^t_{\mathcal{V}}, x^t_{\mathcal{E}})$, where $x^t_{\mathcal{V}} := [x^t_i]_{i=1}^n$ and $x^t_{\mathcal{E}} := [x^t_{ij}]_{i<j}$ by independently sampling each node and edge.

$$x^t_i \sim \text{Cat}\left((1-t)\,\phi_{\mathcal{V}}(x^0_i) + t\,\phi_{\mathcal{V}}(x^{\text{data}}_i)\right), \quad x^t_{ij} \sim \text{Cat}\left((1-t)\,\phi_{\mathcal{E}}(x^0_{ij}) + t\,\phi_{\mathcal{E}}(x^{\text{data}}_{ij})\right). \tag{3.13}$$

where Cat denotes categorical sampling. Let $\hat{x}^t$ be the continuous embedding of the discrete graph x^t via $\phi_{\mathcal{V}}, \phi_{\mathcal{E}}$ (Section 3). While this embedding resides in a continuous ambient space, training is conducted solely on discrete graphs $x \in \mathcal{X}$, as sampling proposals are concerning discrete jumps. The continuous embedding thus mainly facilitates gradient computation of $\nabla_{\hat{x}_t} V_\theta(\hat{x}^t)$.

Objectives. We use the optimality condition derived in Section 3.1 to construct a flow-like loss $\mathcal{L}_{\text{Flow}}$. Specifically, the relation $\nabla_{\hat{y}} \tilde{V}_\theta(\hat{y}) = -\frac{1}{2\eta} \nabla_{\hat{y}} \tilde{c}(\hat{x}, \hat{y})$ from (3.6), evaluated at $\hat{y} = \tilde{T}_k(\hat{x})$, motivates training the negative energy gradient to match the displacement vector v pointing towards the data, since for $\eta = 1$ and $\lambda_{\mathcal{V}} = \lambda_{\mathcal{E}} = 1$, $\frac{1}{2\eta} \nabla_{\hat{y}} \tilde{c}_{\text{loc}}(\hat{x}^0, \hat{y})\Big|_{\hat{y}=\hat{x}^{\text{data}}} = \hat{x}^{\text{data}} - \hat{x}^0 = v$. The training objective minimizes the cost (c_{loc}) along the interpolation path:

$$\mathcal{L}_{\text{Flow}}(\theta) = \mathbb{E}_{(x^0, x^{\text{data}}) \sim \tilde{\gamma},\, t \sim \mathcal{U}([0,1])} \left[\|\nabla_{\hat{x}_t} \tilde{V}_\theta(\hat{x}^t) + v\|_2^2 \right], \tag{3.14}$$

$$\mathcal{L}_{\text{CL}}(\theta) = \mathbb{E}_{x^+ \sim \pi_{\text{data}}}[\tilde{V}_\theta(\hat{x}^+)] - \mathbb{E}_{x^- \sim \text{sg}(\pi_\theta)}[\tilde{V}_\theta(\hat{x}^-)]. \tag{3.15}$$

The contrastive loss $\mathcal{L}_{\text{CL}}$ Hinton [2002] (see Appendix I for the derivation) leverages the discrete proposals introduced in Section 3.1 and Section 3.2 to efficiently sample from the model distribution. The main objective is to minimize $\min_\theta \; \mathcal{L}_{\text{Flow}}(\theta) + \lambda_{\text{CL}}\, \mathcal{L}_{\text{CL}}(\theta)$. We perform a warm-up phase of N_{warmup} training iterations using only $\mathcal{L}_{\text{Flow}}$ before introducing $\mathcal{L}_{\text{CL}}$. This yields higher-quality negatives for contrastive learning, consistent with the continuous formulation [Balcerak et al., 2025]. After warm-up, we temporarily freeze the network and tune the mixing-phase sampler hyperparameters $(\beta_{\text{mh}}, \beta^L, \lambda^L_{\mathcal{V}}, \lambda^L_{\mathcal{E}})$ by minimizing sample energy under fixed V_θ, equivalently maximizing relative likelihood. To approximate samples from π_θ required by $\mathcal{L}_{\text{CL}}$, we run Markov chains initialized in equal proportions from uniform noise and data samples. (See Appendix F for detailed hyperparameters.) Algorithms 1 and 2 summarize training and sampling.

The contrastive formulation can be generalized to allow certain negative samples to be suppressed more strongly than others. We explore this possibility for invalid molecular graphs, with details and an ablation in Appendix F and Table 5.

Algorithm 1 GEM Training

Input: OT Solver, $N_{\text{CL}}, N_{\text{warmup}}, \lambda_{\text{CL}}$
for $i = 0, 1, 2, \ldots$ until V_θ converges **do**
  Sample $(x^0, x^{\text{data}}) \sim \tilde{\gamma}$ (via OT Solver)
  $t \sim \mathcal{U}(0,1)$, $x^t \sim \text{Cat}\left((1-t)\hat{x}^0 + t\hat{x}^{\text{data}}\right)$
  $\mathcal{L}_{\text{Flow}} = \|\nabla \tilde{V}_\theta(\hat{x}^t) + (\hat{x}^{\text{data}} - \hat{x}^0)\|^2$
  if $i > N_{\text{warmup}}$ **then**
    $x \sim$ GEM Sampling$(V_\theta, N_{\text{CL}})$
    $\mathcal{L}_{\text{CL}} = \mathbb{E}[\tilde{V}_\theta(\hat{x}^{\text{data}})] - \mathbb{E}[\tilde{V}_\theta(\hat{x})]$
  else $\mathcal{L}_{\text{CL}} = 0$ **end if**
  optimizer_step$(\mathcal{L}_{\text{Flow}} + \lambda_{\text{CL}}\mathcal{L}_{\text{CL}})$
end for

Algorithm 2 GEM Sampling

Input: V_θ, N_{CL}
$V_{\text{th}} = \mathbb{E}_{\pi_{\text{data}}}[V_\theta]$
Init. $x \sim \pi_0$ (or π_{data})
while $V_\theta(x) > V_{\text{th}}$ **and** x not stuck **do**
  $x \sim q_{\text{greedy}}(x \to y)$
end while
for $k = 1$ **to** N_{CL} **do**
  $x \sim$ MH-step$(q_{\text{mixing}}(x \to y))$
end for
Return x

(a) MOSES (∼1.5M molecules).

Method	V.U.N. ↑	FCD ↓
Training samples (MOSES)	0.0	0.25
Noise Initialization		
GraphEBM* [Liu et al., 2021]	0.081	9.83
VFM* [Eijkelboom et al., 2024]	0.814	2.71
DeFoG* [Qin et al., 2025]	0.822	1.95
GEM (Ours)	**0.923**	**1.39**
Data Initialization		
GraphEBM* [Liu et al., 2021]	0.322	7.31
GEM (Ours)	**0.898**	**0.76**

(b) QM9 (∼0.15M molecules).

Method	V.U.N. ↑	FCD ↓
Training samples (QM9)	0.000	0.04
Noise Initialization		
GraphEBM* [Liu et al., 2021]	0.079	7.16
VFM [Eijkelboom et al., 2024]	0.489	**0.44**
DeFoG [Qin et al., 2025]	0.348	0.81
GEM (Ours)	**0.612**	0.92
Data Initialization		
GraphEBM* [Liu et al., 2021]	0.103	5.15
GEM (Ours)	**0.502**	**0.31**

Table 1: **Unconditional Generation.** Results by initialization for (a) MOSES and (b) QM9. Both report V.U.N. and FCD; higher V.U.N. and lower FCD are better. *Matched DeFoG backbone; EBMs use a small energy head (Appendix A). MOSES: 25k generations, 1000 steps; QM9: 10k generations, 500 steps.

4 Experiments

We use two established molecular datasets, QM9 Ramakrishnan et al. [2014] and MOSES Polykovskiy et al. [2020]. QM9 comprises ∼150K small organic molecules containing up to nine heavy atoms (excluding hydrogens), whereas MOSES is a larger, drug-like dataset with ∼1.5M molecules featuring up to 27 heavy atoms. We report (i) generation on both QM9 and MOSES, (ii) conditional generation (property optimization) on MOSES, and (iii) analyses of graph-to-graph geodesics on MOSES. We additionally evaluate GEM on the non-molecular Planar and SBM datasets [Martinkus et al., 2022] (Appendix L).

For unconditional generation on QM9 and MOSES, we report valid-unique-novel (V.U.N.) and Fréchet ChemNet Distance (FCD) [Preuer et al., 2018]. For conditional tasks, we report conditional valid-unique-novel (C.V.U.N.), measuring the proportion of generated graphs that are valid, unique, novel, and fulfill prescribed property conditions. We evaluate two initialization regimes: noise initialization, starting from easy-to-sample noise and targeting valid molecules, and data initialization, natural for EBMs, starting from training molecules and targeting novel edits. Baseline details, experimental setup, uncertainty estimates, hyperparameter sensitivity, and wall-clock measurements are provided in Appendices E and F.

4.1 Unconditional Generation

MOSES. Table 1 (left) reports MOSES results. Across both noise and data initialization, GEM outperforms baselines in terms of V.U.N. and FCD. Figure 4 contrasts performance versus inference steps and initialization strategies, respectively. At larger inference budgets, GEM substantially surpasses diffusion, yielding notably higher V.U.N. and improved distributional fidelity (lower FCD). Wall-clock measurements show the same qualitative trend (Table 7). The ablation results in Table 6 show that the transport-aligned phase, although an approximation of the optimal transport, is crucial in practice to move chains toward data-like high-probability regions. See Appendix K for additional baselines.

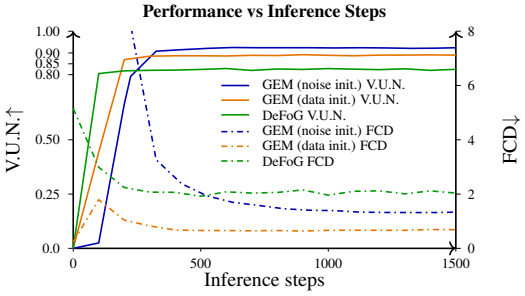

Figure 4: **V.U.N. and FCD vs Steps.** V.U.N. (higher is better) and FCD (lower is better) versus inference steps on MOSES for GEM noise initialization (uniform) with greedy warmup proposal, GEM data initialization with annealed proposal, and DeFoG (marginal).

QM9. Table 1 (right) shows results on QM9. Because QM9 is small, generalizing to novel samples is difficult, and FCD can be affected by novelty variance. GEM achieves the highest V.U.N. under both initializations and the best FCD under data initialization. Under noise initialization, GEM has

MOSES (∼1.5M molecules)

Method	Condition: logS ≥ -2.25		Condition: QED ≥ 0.9		Condition: TPSA ≤ 50		Condition: logP ≤ 1.5	
	C.V.U.N. ↑	C.V.U. ↑	C.V.U.N. ↑	C.V.U. ↑	C.V.U.N. ↑	C.V.U. ↑	C.V.U.N. ↑	C.V.U. ↑
Training samples (MOSES)	0.0	0.24	0.0	0.16	0.0	0.21	0.0	0.15
Noise Initialization								
G-VFM* [Eijkelboom et al., 2025]	0.72	0.75	0.26	0.28	0.72	0.80	0.70	0.74
DeFoG* [Qin et al., 2025]	0.74	0.79	0.26	0.31	0.77	**0.82**	0.78	0.82
GEM (Ours)	**0.80**	**0.81**	**0.40**	**0.40**	**0.80**	0.81	**0.85**	**0.85**

Table 2: **Property Optimization.** MOSES conditional generation under property constraints. C.V.U.N. is the fraction meeting the condition and valid, unique, novel; conditional valid-unique (C.V.U.) is the fraction meeting the condition and valid, unique. 1k inference steps, 5k samples. * Reproduced with matched backbones (Appendix A).

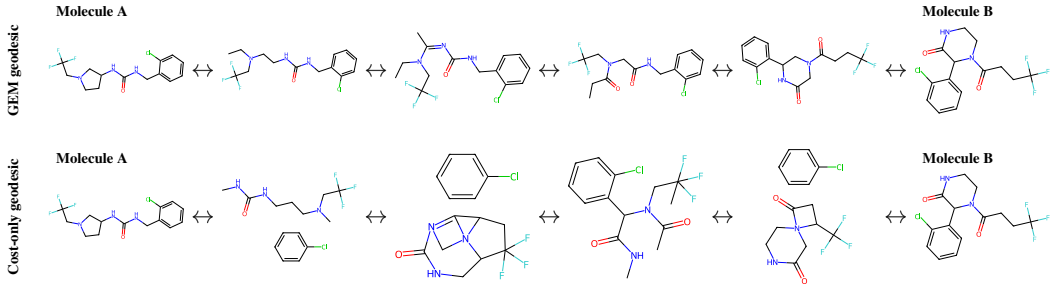

Figure 5: **Qualitative Geodesic Paths.** Representative MOSES interpolation from Molecule A to Molecule B. Columns show successive samples along each path; GEM uses the learned energy-weighted geometry, while the baseline uses only graph-edit cost.

higher novelty than VFM (0.63 vs. 0.49), a tradeoff reflected in higher FCD. Individual metrics are reported in Appendix K.

4.2 Conditional Generation

We optimize molecular properties (logP, logS, quantitative estimate of drug-likeness (QED), and topological polar surface area (TPSA)) by following [Vignac et al., 2023, Ninniri et al., 2025]: we train a regressor f_ϕ and sample from $V_\theta^{\text{cond}}(x) = V_\theta(x) + \lambda_{\text{prop}}\|f_\phi(x) - \zeta\|^2$. We run the same proposal kernel using $-\nabla_{\hat{x}} V_\theta^{\text{cond}}(\hat{x})$, with the MH ratio modified accordingly. For fairness, all methods in Table 2 use noise-conditioned regressors and comparable tuning; thresholds and implementation details are in Appendices F and G. We observe that GEM maintains strong validity, uniqueness, and novelty while improving constraint satisfaction relative to baselines (Table 2).

4.3 Geodesic Analysis

We study molecule-to-molecule interpolation by computing geodesics (low-energy graph-edit paths aligned with the learned data geometry [Béthune et al., 2025]). We report average validity versus the endpoint distance with distance computed via the node-matching approximation to c_{FGW} (Appendix C). Figure 5 shows qualitative paths. As seen in Figure 6, energy-weighted geodesics preserve validity more reliably than cost-only paths, indicating that the energy encodes chemically meaningful structure. See Appendix H for experimental details.

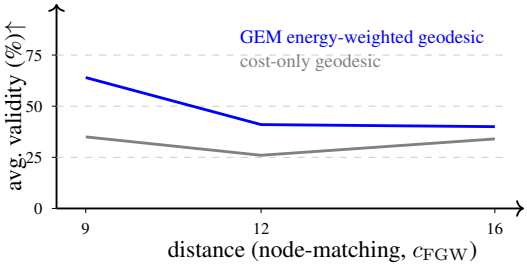

Figure 6: **Validity Along Geodesics.** Average chemical validity (%) vs. distance (node-matching, c_{FGW}) for GEM energy-weighted versus cost-only geodesics. Cost-only ignores the data geometry and interpolates through low-validity regions.

5 Discussion and Conclusion

We introduced *Graph Energy Matching* (GEM), a discrete generative framework that extends continuous energy matching principles to discrete graph domains motivated by the transport-map formulation

of the JKO scheme. GEM integrates deterministic, transport-aligned graph edits with effective mixing under a unified scalar energy model. The transport stage serves primarily as an effective burn-in mechanism that guides sampling toward high-probability regions. Conceptually, this allows (i) scalable and high-quality molecular graph generation from both noise- and data-initialized states, and (ii) flexible conditional generation at inference-time through an explicitly learned relative-likelihood structure. Empirically, GEM achieves state-of-the-art performance among graph EBMs, matching or surpassing discrete diffusion baselines on most reported MOSES and QM9 metrics. While our empirical evaluation focuses primarily on molecular graphs, GEM also remains highly competitive on the non-molecular Planar and SBM datasets. Further, it effectively addresses property optimization under threshold constraints. Moreover, GEM's representation naturally enables diagnostics such as graph geodesics.

Broader Context and Future Directions. Our current formulation operates in a one-hot embedding space, where distinct categories at each node or edge are equally separated. Although the learned energy distinguishes between edits, the embedding geometry itself does not encode similarities between categories. Learning more compact representations that capture these similarities could provide a more informative geometry for transport and proposal construction. More broadly, graphs are a particular class of structured discrete objects. Extending GEM's transport–mixing formulation to other discrete object distributions would require suitable representations, edit operations, and transport costs. Exploring how the same scalar energy can support transport and mixing across these domains is an important direction for future work.

Limitations. GEM learns a scalar energy field capturing relative likelihood. Consequently, like other EBMs, generation velocities must be derived from gradients, incurring a $\sim$2–3$\times$ theoretical overhead compared to purely forward parameterizations.

Acknowledgments

We thank Antonio Terpin for helpful discussions. This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence. Support came from the Helmut Horten Foundation and the European Cooperation in Science and Technology (COST). We declare no competing interests.

References

Michael Samuel Albergo, Mark Goldstein, Nicholas Matthew Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *International Conference on Machine Learning*, pages 921–937. PMLR, 2024.

Michal Balcerak, Tamaz Amiranashvili, Antonio Terpin, Suprosanna Shit, Lea Bogensperger, Sebastian Kaltenbach, Petros Koumoutsakos, and Bjoern Menze. Energy matching: Unifying flow matching and energy-based models for generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, pages 8583–8609, 2025.

Eric W. Bell and Yang Zhang. DockRMSD: An open-source tool for atom mapping and RMSD calculation of symmetric molecules through graph isomorphism. *Journal of Cheminformatics*, 11:40, 2019.

Louis Béthune, David Vigouroux, Yilun Du, Rufin VanRullen, Thomas Serre, and Victor Boutin. Follow the energy, find the path: Riemannian metrics from energy-based models. In *Advances in Neural Information Processing Systems*, volume 38, pages 97824–97870, 2025.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *International Conference on Machine Learning*, pages 5453–5512. PMLR, 2024.

Yilun Du. *Learning Generalizable Systems by Learning Composable Energy Landscapes*. PhD thesis, Massachusetts Institute of Technology, 2025.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, volume 32, pages 3608–3618, 2019.

Floor Eijkelboom, Grigory Bartosh, Christian A. Naesseth, Max Welling, and Jan-Willem van de Meent. Variational flow matching for graph generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 11735–11764, 2024.

Floor Eijkelboom, Heiko Zimmermann, Sharvaree Vadgama, Erik J. Bekkers, Max Welling, Christian A. Naesseth, and Jan-Willem van de Meent. Controlled generation with equivariant variational flow matching. In *International Conference on Machine Learning*, pages 15066–15078. PMLR, 2025.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Dominik Fuchsgruber, Tom Wollschläger, and Stephan Günnemann. Energy-based epistemic uncertainty for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 37, pages 34378–34428, 2024.

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris Maddison. Oops I took a gradient: Scalable sampling for discrete distributions. In *International Conference on Machine Learning*, pages 3831–3841. PMLR, 2021.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1): 97–109, 1970.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pages 2323–2332. PMLR, 2018.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker–Planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Greg Landrum et al. RDKit: Open-source cheminformatics, 2026. URL https://www.rdkit.org.

Nicolas Lanzetti, Antonio Terpin, and Florian Dörfler. Variational analysis in the Wasserstein space. *arXiv preprint arXiv:2406.10676*, 2024.

Nicolas Lanzetti, Saverio Bolognani, and Florian Dörfler. First-order conditions for optimization in the Wasserstein space. *SIAM Journal on Mathematics of Data Science*, 7(1):274–300, 2025.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. Energy-based models. In *Predicting Structured Data*. MIT Press, 2007.

Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. GraphEBM: Molecular graph generation with energy-based models. *arXiv preprint arXiv:2102.00546*, 2021.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*, pages 23321–23337. PMLR, 2023.

Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. SPECTRE: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In *International Conference on Machine Learning*, pages 15159–15179. PMLR, 2022.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Matteo Ninniri, Marco Podda, and Davide Bacciu. Graph diffusion that can insert and delete. In *Advances in Neural Information Processing Systems*, volume 38, pages 78375–78401, 2025.

Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual Review of Physical Chemistry*, 71(1):361–390, 2020.

Pavel G. Polishchuk, Timur I. Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-Aided Molecular Design*, 27(8):675–679, 2013.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11: 565644, 2020.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. DeFoG: Discrete flow matching for graph generation. In *International Conference on Machine Learning*, pages 50269–50326. PMLR, 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):140022, 2014.

Antoine Siraudin, Fragkiskos D. Malliaros, and Christopher Morris. Cometh: A continuous-time discrete-state graph diffusion model. *Transactions on Machine Learning Research*, 2025.

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackerman, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13, 2020.

Antonio Terpin, Nicolas Lanzetti, Martín Gadea, and Florian Dörfler. Learning diffusion at lightspeed. In *Advances in Neural Information Processing Systems*, volume 37, pages 6797–6832, 2024.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR, 2019.

Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, pages 16506–16527, 2023.

Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. Energy-based out-of-distribution detection for graph neural networks. In *International Conference on Learning Representations*, pages 17290–17306, 2023.

Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi S. Jaakkola. Crystal diffusion variational autoencoder for periodic material generation. In *International Conference on Learning Representations*, pages 7257–7276, 2022.

Giacomo Zanella. Informed proposals for local MCMC in discrete spaces. *Journal of the American Statistical Association*, 115(530):852–865, 2020.

Ruqi Zhang, Xingchao Liu, and Qiang Liu. A Langevin-like sampler for discrete distributions. In *International Conference on Machine Learning*, pages 26375–26396. PMLR, 2022.

A Permutation-invariant energy parameterization

We implement the potential with a graph transformer backbone directly from DeFoG Qin et al. [2025]. Each layer of this backbone is permutation equivariant, meaning that for any permutation (relabeling) σ, the layer satisfies:

$$h^{(l)}(\sigma \cdot x_{\mathcal{V}}^{(l)}, \sigma \cdot x_{\mathcal{E}}^{(l)}, f^{(l)}) = (\sigma \cdot x_{\mathcal{V}}^{(l+1)}, \sigma \cdot x_{\mathcal{E}}^{(l+1)}, f^{(l+1)}),$$

where $x_{\mathcal{V}}^{(l)}$ and $x_{\mathcal{E}}^{(l)}$ denote node and edge features at layer l, respectively, and $f^{(l)}$ denotes the global graph-level feature.

Following [Vignac et al., 2023, Qin et al., 2025], we pad and mask all smaller graphs to the maximum number of nodes observed in the training dataset, denoted as n. This results in fixed-size inputs to the model, denoted $x_{\mathcal{V}}$ and $x_{\mathcal{E}}$ for node and edge features, respectively. We denote node and edge features at layer l as $x_{\mathcal{V}}^{(l)}$ and $x_{\mathcal{E}}^{(l)}$, respectively. The global feature $f^{(l)}$ collects graph-level features at

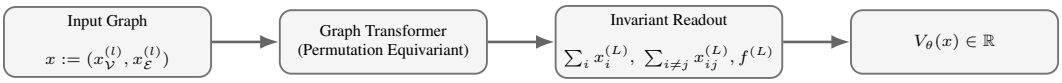

Figure 7: **Energy Backbone.** Permutation-equivariant backbone with invariant pooling to produce a scalar energy.

layer l and is updated through a permutation-invariant operation. Figure 7 summarizes the backbone and invariant readout.

The scalar energy is obtained by an invariant readout that first pools the equivariant features and then applies a multilayer perceptron (MLP):

$$\begin{aligned} \bar{x} = \sum_{i\in\mathcal{V}} x_i^{(L)}, \qquad \bar{e} = \sum_{i<j} x_{ij}^{(L)}, \\ V_\theta(x) = \mathrm{MLP}\big([\bar{x}, \bar{e}, f^{(L)}]\big). \end{aligned} \tag{A.1}$$

Since the sum operation is permutation invariant, the concatenated vector $(\bar{x}, \bar{e}, f^{(L)})$ is invariant. Therefore, an MLP applied to it is also invariant.

Energy head and parameter count. To enhance expressivity, we incorporate Relative Random Walk Probabilities (RRWP) [Ma et al., 2023] as node and edge features, matching DeFoG for a fair comparison. The energy MLP head is a 2-layer MLP that maps the pooled hidden graph representation (concatenated node/edge/f embeddings) to a single scalar energy. With dimensions $448 \to 256 \to 1$ and SiLU activation, it has 115,201 parameters. The full model has 16,382,158 trainable parameters (≈ 16.38M).

B JKO reduction to a transport map

We specialize the continuous JKO functional (2.1) to discrete distributions in the transport regime ($\epsilon = 0$) and replace the time step Δt with the jump size η. Given π_k and a candidate π, the variational objective reads

$$\begin{aligned} \pi_{k+1} = \arg\min_{\pi} \frac{1}{2\eta} \inf_{\gamma\in\Gamma(\pi_k,\pi)} \mathbb{E}_{(x,y)\sim\gamma}[c(x,y)] \\ + \mathbb{E}_{y\sim\pi}[V_\theta(y)]. \end{aligned} \tag{B.1}$$

Assuming the optimal coupling admits a deterministic transport map T (i.e., a Monge-type solution) so that $\pi = T_{\#}\pi_k$, the objective reduces to an optimization over maps:

$$\begin{aligned} T \in \arg\min_{T':\pi=T'_{\#}\pi_k} \frac{1}{2\eta} \mathbb{E}_{x\sim\pi_k}[\, c(x, T'(x))\,] \\ + \mathbb{E}_{x\sim\pi_k}[\, V_\theta(T'(x))\,]. \end{aligned} \tag{B.2}$$

$$T \in \arg\min_{T'} \mathbb{E}_{x\sim\pi_k} \left[\frac{1}{2\eta} c(x, T'(x)) + V_\theta(T'(x)) \right]. \tag{B.3}$$

Since the expectation is over $x \sim \pi_k$, the minimization decouples pointwise, yielding

$$T(x) \in \arg\min_{y} \left\{ \frac{1}{2\eta} c(x,y) + V_\theta(y) \right\}, \quad \text{for each } x \sim \pi_k. \tag{B.4}$$

This yields the update used in Section 3.1.

Scope of the transport interpretation. This derivation is used as modeling motivation for the training objective and transport-aligned proposals. It does not imply exact optimal transport for the complete implemented sampler, which uses discrete graph edits, approximate minibatch matching, and local linearization. We therefore do not claim end-to-end formal optimality for the practical sampler.

C Permutation-invariant minibatch matching

Exact computation of the hard-permutation FGW cost c_{FGW} requires a combinatorial search over permutations. We therefore use fast, permutation-invariant hard-assignment approximations tailored to the source distribution and the downstream use (pairing vs. local edits).

Histogram matching for a uniform source. When the source π_0 is a uniform distribution over node and edge classes, we use a cheap signature for each graph. Let $h_{\mathcal{V}}$ be the normalized node-type histogram, $h_{\mathcal{E}}$ the normalized edge-type histogram (over $i < j$), and $h_{\mathcal{VE}}$ the histogram over unordered node-type pairs with edge types. We form a weighted signature $h(x) = [\alpha_1 h_{\mathcal{V}},\ \alpha_2 h_{\mathcal{E}},\ \alpha_3 h_{\mathcal{VE}}]$ and solve a linear assignment between graphs of equal size using $\|h(x) - h(y)\|_1$. This produces permutation-invariant couplings and is sufficient when π_0 is far from π_{data}, so we do not require fine-grained alignment. α_* are chosen to balance each contribution equally.

Node matching for molecule-like sources. If the source distribution already produces valid molecules, we use a node-matching permutation alignment based on node labels, followed by computing the aligned local cost. This reflects chemical atom mapping practice: once atoms are aligned, bond correspondences are largely determined [Bell and Zhang, 2019].

D Sampling proposal implementation details

Gradient features. We use gradient-informed proposals on the discrete graph space with target density $\pi_\theta(x) \propto \exp\left(-\beta_{\mathrm{mh}} V_\theta(x)\right)$. To compute the gradients of the potential, we first embed x into a continuous, real-valued one-hot representation $\hat{x}$, and then evaluate the gradient with respect to this embedding as $g := \nabla_{\hat{x}} V_\theta(\hat{x})$. We reshape node and edge gradients as

$$g^{\mathcal{V}} := \mathrm{reshape}_{n, l_{\mathrm{node}}}\left(\nabla_{\hat{x}_{\mathcal{V}}} V_\theta(\hat{x})\right), \tag{D.1}$$

$$g^{\mathcal{E}} := \mathrm{reshape}_{\binom{n}{2}, l_{\mathrm{edge}}}\left(\nabla_{\hat{x}_{\mathcal{E}}} V_\theta(\hat{x})\right), \tag{D.2}$$

where $\mathrm{reshape}_{m,n}(\cdot)$ is a row-major reshape into $\mathbb{R}^{m \times n}$.

Since we are working with undirected graphs, we use symmetrized edge gradients from the graph transformer.

Greedy transport implementation. One forward pass and one backward pass compute the energy gradient used to score all categorical node and edge replacements in (3.7). We retain the best replacement at each site, rank these candidates jointly, and apply the top N edits to construct one proposed graph without enumerating $C_N(x)$. We use $N = 1$ for the main QM9 and MOSES experiments. The parameter N controls only transport and does not fix the number of edits during mixing. See Appendix F.3 for sensitivity to N and computational costs.

Factorized proposal (multi-site). We sample all nodes and undirected edges independently from categorical logits

$$\ell^{\mathcal{V}}_{i,c_i} = \beta\left(g^{\mathcal{V}}_{i,x_i} - g^{\mathcal{V}}_{i,c_i}\right) - \lambda_{\mathcal{V}}\, \mathbb{I}_{\{c_i \neq x_i\}};\ \forall\, c_i \in \mathcal{X}_{\mathrm{node}} \tag{D.3}$$

$$\ell^{\mathcal{E}}_{ij,c_{ij}} = \beta\left(g^{\mathcal{E}}_{ij,x_{ij}} - g^{\mathcal{E}}_{ij,c_{ij}}\right) - \lambda_{\mathcal{E}}\, \mathbb{I}_{\{c_{ij} \neq x_{ij}\}};\ \forall\, c_{ij} \in \mathcal{X}_{\mathrm{edge}} \tag{D.4}$$

so the full proposal factorizes as

$$q_\beta(x \to y) = \prod_{i \in \mathcal{V}} \mathrm{Cat}\left(y_i; \mathrm{softmax}(\ell^{\mathcal{V}}_i)\right) \times \prod_{i<j} \mathrm{Cat}\left(y_{ij}; \mathrm{softmax}(\ell^{\mathcal{E}}_{ij})\right). \tag{D.5}$$

$\lambda_{\mathcal{V}}$ and $\lambda_{\mathcal{E}}$ penalize modifications to nodes and edges, respectively.

MH refinement (single beta). For the fixed-temperature proposal we set $\beta^L = \beta_{\text{mh}} = \beta$ and accept

$$\begin{aligned} \alpha(x, y) &= \min\{1, \exp(\Delta(x, y))\}, \\ \Delta(x, y) &= -\beta_{\text{mh}}\big(V_\theta(y) - V_\theta(x)\big) + \log \frac{q_{\beta^L}(y \to x)}{q_{\beta^L}(x \to y)}. \end{aligned} \tag{D.6}$$

Guarantee scope. The formal Markov-chain guarantee applies to the fixed-temperature MH mixing stage: with the acceptance ratio above, the chain satisfies detailed balance with target π_θ, so π_θ is stationary. The preceding transport stage and optional temperature annealing are practical burn-in mechanisms that accelerate entry into high-probability regions before stationary mixing.

Two-betas simulated annealing. We decouple the proposal temperature and target temperature by using β^L in the logits above, while β_{mh} controls the MH ratio. Optionally, β_{mh} is annealed across S_{anneal} mixing steps:

$$\beta_{\text{mh}}(s) = \begin{cases} \beta_{\text{mh}}^{\text{init}} + \Delta\beta \, \dfrac{s}{S_{\text{anneal}} - 1}, & 0 \le s < S_{\text{anneal}}, \\ \beta_{\text{mh}}^{\text{final}}, & s \ge S_{\text{anneal}}. \end{cases} \tag{D.7}$$

where $\Delta\beta = \beta_{\text{mh}}^{\text{final}} - \beta_{\text{mh}}^{\text{init}}$.

Do-nothing resampling and proposal diffusion. If the independently sampled proposal equals the current state (all sites *stay*), we resample after softening the logits to increase randomness: $\beta^L \leftarrow \rho\,\beta^L$, $\lambda_{\mathcal{V}} \leftarrow \rho\,\lambda_{\mathcal{V}}$, $\lambda_{\mathcal{E}} \leftarrow \rho\,\lambda_{\mathcal{E}}$ with $0 < \rho < 1$. We repeat this a bounded number of times; if no change occurs, we keep the current state and accumulate the *stay* probability in $\log q$. The reverse transition uses the same resampling rule, preserving the MH ratio.

E Conceptualizing baselines

We summarize the baseline methods used in our experiments with an emphasis on the generation/sampling mechanism for each method.

GraphEBM [Liu et al., 2021] models molecular graphs using an energy-based model that assigns a scalar energy $E_\theta(x)$ to each graph x via a graph neural network. Unlike transport- or flow-based approaches, GraphEBM does not learn an explicit transformation from a source distribution to the data distribution. Instead, generation depends exclusively on Langevin dynamics sampling within a continuous embedding space.

DeFoG [Qin et al., 2025] adapts discrete flow matching to graphs by pairing an explicit noising path (via linear interpolation toward a simple base distribution) with a learned CTMC denoiser: a network predicts the clean-graph posterior from an intermediate noisy graph, which in turn defines the time-dependent rate matrices used for generation. Sampling starts from the base distribution and simulates the resulting CTMC toward the data distribution, with the sampling step schedule largely selectable at inference time.

VFM [Eijkelboom et al., 2024] casts flow matching as variational inference over trajectory endpoints (the "posterior probability path"), yielding a KL-based objective that for categorical graph variables reduces to a per-component cross-entropy. In contrast to DeFoG's stochastic CTMC jump dynamics, VFM targets a deterministic continuous flow (vector field) on the probability simplex and generates by integrating this flow from the base distribution, then discretizing/sampling from the final categorical probabilities.

G-VFM [Eijkelboom et al., 2025] extends VFM with equivariant conditioning mechanisms to support controllable generation.

F Experimental setup and hyperparameters

MOSES (shared across experiments). Training is conducted in two stages. We first train with $\lambda_{\text{CL}} = 0.0$ for 330 k iterations (lr $= 10^{-4}$, batch size $= 128$) to pretrain the potential field, enabling efficient burn-in of samples initialized from noise. We then follow with $\lambda_{\text{CL}} = 0.1$ for 1000 iterations using chain length N_{CL} = 500 at lr $= 1 \times 10^{-5}$, batch size 128. Greedy transport proposals followed by mixing with $\beta^L = 9.55$, $\lambda_{\mathcal{V}}^L = 0.23$, $\lambda_{\mathcal{E}}^L = 1.88$, and $\beta_{\text{mh}} = \beta^L$. For the DeFoG baselines, we directly use the authors' provided checkpoint [Qin et al., 2025]. During training, we consider two distinct setups for initializing the contrastive loss (CL) samples: either from noise or directly from data. We evenly split these initializations (50% noise / 50% data), except in the unconditional generation scenario ("Data Initialization" category), where we exclusively (100%) initialize CL samples from data.

Validity-aware contrastive weighting. We introduce validity-aware contrastive weighting controlled by a single hyperparameter, λ_{invalid}. For each training graph, we run one chain, initialized from that graph or from noise with the same number of nodes. The training graph and the chain's final sample form the contrastive pair. The sums below group these minibatch pairs (x^+, x^-) by whether the generated negative passes RDKit [Landrum et al., 2026] sanitization and is connected (valid), or fails either check (invalid):

$$\mathcal{L}_{\text{CL}} = \frac{\sum_{\text{valid}} \left[V_\theta(x^+) - V_\theta(x^-)\right] + \lambda_{\text{invalid}} \sum_{\text{invalid}} \left[V_\theta(x^+) - V_\theta(x^-)\right]}{\sum_{\text{valid}} 1 + \lambda_{\text{invalid}} \sum_{\text{invalid}} 1}.$$

For $\lambda_{\text{invalid}} > 1$, invalid negatives receive greater weight than valid ones, encouraging higher energy relative to their paired data graphs. This validity-aware component of GEM uses the flexibility of energy-based models to selectively suppress undesired samples during training. We use $\lambda_{\text{invalid}} = 10$ for noise-initialized MOSES generation (Table 1a) and $\lambda_{\text{invalid}} = 1$ for QM9. Around 99% of generated QM9 samples are already valid, leaving limited room for validity improvements through additional weighting. Table 5 reports the MOSES weighting ablation.

F.1 Unconditional generation

We report 25k MOSES samples and 10k QM9 samples. For noise initialization we use greedy transport followed by the mixing proposal with the parameters from the training setup. For data initialization we use an annealed schedule with $\beta^L = 8.12$, $\beta_{\text{mh}}^{\text{init}} = 0.18$, $\beta_{\text{mh}}^{\text{final}} = 13.56$, $S_{\text{anneal}} = 200$, $\lambda_{\mathcal{V}}^L = 0.07$, and $\lambda_{\mathcal{E}}^L = 2.23$.

F.2 Conditional generation and property optimization

Noisy property regressors. We train time-conditioned regressors on MOSES using shared hyperparameters for both marginal- and uniform-noise variants. Training uses batch size 1024 for 60 epochs with Adam (lr $= 10^{-4}$, weight decay=10^{-6}). The regressor is a GraphTransformer with 4 layers, 8 attention heads, hidden dimensions $(d_x, d_e, d_y) = (256, 128, 128)$, feedforward dimensions $(256, 128, 192)$, and MLP dimensions $(X, E, y) = (256, 128, 256)$, SiLU activations, and a scalar property head. We apply polydec time distortion. Additional noise-sensitivity details are provided in Appendix G.

Conditional generation (GEM & DeFoG). For a desired property value ζ, we define $V_\theta^{\text{cond}}(x) = V_\theta(x) + \lambda_{\text{prop}} \|f_\phi(x) - \zeta\|^2$ and run the same proposal kernel using the gradient of V_θ^{cond}, with the MH acceptance ratio modified accordingly. For all methods in Table 2, we use noise-conditioned regressors and comparable hyperparameter tuning. We feed regressor time using an energy-based proxy that increases linearly until the chain reaches the data-distribution energy level, then clamp it to $t = 1$ (Appendix G). The target constraints (logS $\geq$ -2.25, logP $\leq$ 1.5, QED $\geq$ 0.9, TPSA $\leq$ 50) are selected so that unconditional MOSES samples satisfy each constraint approximately 20% of the time.

GEM mixing proposal parameters: ($\beta^L = 9.55$, $\lambda_{\mathcal{V}}^L = 0.23$, $\lambda_{\mathcal{E}}^L = 1.88$) with uniform noise initialization/regressors. DeFoG uses marginal-noise initialization/ regressors. For each task we sweep λ_{prop} and report the best CVUN.

Task	GEM λ_{prop}	DeFoG λ_{prop}
logP ≤ 1.5	10	140
logS ≥ -2.25	16	130
QED ≥ 0.9	193	360
TPSA ≤ 50	0.03	0.5

Table 3: Hyper-parameters for noisy-regressor conditioning on MOSES. GEM uses uniform-noise regressors; DeFoG uses marginal-noise regressors. Conditional results based on 5k samples.

The reported results are computed directly using RDKit library Landrum et al. [2026]—not from the regressors, which serve solely for guidance purposes.

F.3 Robustness, ablations, sensitivity, and runtime

Multi-seed variability. We conducted additional multi-seed experiments with $\lambda_{\text{invalid}} = 1$, distinguishing training-seed variability from sampling-seed variability. On MOSES, training across five seeds yielded V.U.N. $= 0.858 \pm 0.013$ and FCD $= 1.51 \pm 0.03$. Sampling from a fixed checkpoint across five random seeds yielded V.U.N. $= 0.855 \pm 0.011$ and FCD $= 1.51 \pm 0.03$.

Source of variability	V.U.N. ↑	FCD ↓
Training seeds	0.858 ± 0.013	1.51 ± 0.03
Sampling seeds, fixed checkpoint	0.855 ± 0.011	1.51 ± 0.03

Table 4: MOSES unconditional variability across five seeds.

For conditional settings, C.V.U.N. (logS/QED/TPSA/logP) showed training-seed variability of 0.80 ± 0.02 / 0.40 ± 0.02 / 0.80 ± 0.02 / 0.85 ± 0.02. With a fixed checkpoint, sampling-seed standard deviations were 0.01 / 0.02 / 0.02 / 0.02, respectively. These results indicate low variance across both training and sampling randomness.

Invalid-negative weighting ablation. Table 5 compares noise-initialized MOSES generation at three values of λ_{invalid}. Increasing it from 1 to 10 improves both V.U.N. and FCD. Increasing it further to 100 has little effect.

λ_{invalid}	V.U.N. ↑	FCD ↓
1	0.856	1.51
10	0.923	1.39
100	0.921	1.40

Table 5: **Invalid-negative weighting.** Noise-initialized GEM generation on MOSES.

Transport/mixing ablation. We isolate the transport phase, mixing phase, and their switching combination on MOSES with $\lambda_{\text{invalid}} = 1$ under noise initialization with 1000 inference steps. Using the same checkpoint, full GEM reaches V.U.N. $= 0.856$, while transport-only reaches 0.712 and mixing-only reaches 0.485. Retraining with only mixing further degrades results because the lack of burn-in destabilizes contrastive learning, leading to early collapse after roughly 60 iterations and V.U.N. $= 0.21$.

Variant	V.U.N. ↑
Full GEM (transport + mixing)	**0.856**
Transport-only, same checkpoint	0.712
Mixing-only, same checkpoint	0.485
Mixing-only, retrained	0.21

Table 6: Transport/mixing ablation on MOSES under noise initialization.

Sampling hyperparameter sensitivity. Sampling-stage hyperparameters $(\beta_{\mathrm{mh}}, \beta^L, \lambda_{\mathcal{V}}^L, \lambda_{\mathcal{E}}^L)$ were tuned to minimize sample energy under fixed V_θ. Evaluations on MOSES with $\lambda_{\mathrm{invalid}} = 1$ over 100 independent tuning runs, each with 256 samples, yielded stable V.U.N. results: 0.856, 0.861, and 0.848 for the three best configurations. This indicates limited sensitivity to the final sampling hyperparameter choice.

Wall-clock runtime. We additionally report MOSES wall-clock measurements on an NVIDIA H100 with batch size 128, using $\lambda_{\mathrm{invalid}} = 1$ for GEM. Consistent with Figure 4, DeFoG improves rapidly at early times, while GEM reaches stronger final V.U.N. and FCD at higher computational budgets.

Time (s)	0	1	2	3	6	9	12	15
V.U.N. (DeFoG)	0.000	0.788	0.823	0.820	0.821	0.821	0.823	0.825
V.U.N. (Data-init)	0.000	0.234	0.467	0.692	0.885	0.887	0.887	0.891
V.U.N. (Noise-init)	0.000	0.012	0.056	0.370	0.836	0.855	0.861	0.860
FCD (DeFoG)	5.150	2.495	1.956	2.039	2.097	2.016	2.021	2.036
FCD (Data-init)	0.244	1.060	1.748	1.353	0.799	0.662	0.653	0.655
FCD (Noise-init)	26.16	19.41	16.01	12.61	4.427	2.596	1.895	1.714

Table 7: MOSES wall-clock generation results on an NVIDIA H100, batch size 128.

Sensitivity to the number of transport edits. We vary N using a fixed contrastively trained MOSES checkpoint, categorical-noise initialization, and 5,000 samples per setting with 200- or 500-step chains. Table 8 shows similar V.U.N. for $N = 1$, 5, and 10, while $N = 100$ substantially reduces validity and V.U.N. scores.

	200 steps			500 steps		
N	Validity ↑	Novelty ↑	V.U.N. ↑	Validity ↑	Novelty ↑	V.U.N. ↑
1	0.7566	0.9831	0.7438	0.8896	0.9676	0.8605
5	0.7664	0.9838	0.7540	0.8891	0.9635	0.8560
10	0.7582	0.9855	0.7472	0.8836	0.9631	0.8505
100	0.4592	0.9993	0.2862	0.5928	0.9995	0.4078

Table 8: **Sensitivity to the number of transport edits.** Noise-initialized MOSES generation. All metrics are fractions.

Proposal-ranking cost. Table 9 compares exact energy-difference ranking with gradient scoring for single-edge alternatives. Exact ranking is faster at 16 nodes, but approximately $10\times$ and $159\times$ slower at 32 and 64 nodes, respectively. Exact energy differences are still used in the MH acceptance step.

Nodes	Edge candidates	Exact ranking (ms)	Gradient scoring (ms)
16	120	18.3	29.9
32	496	286.7	28.8
64	2016	4658.2	29.3

Table 9: **Proposal-ranking time.** Single-edge candidates only.

Graph-size scaling. Table 10 reports wall-clock time and peak memory per network evaluation on one NVIDIA H100 with batch size 1. GEM's forward costs are comparable to the baselines, while gradient computation includes an additional backward pass and increases runtime and memory use. The dense edge representation is the main memory bottleneck.

Nodes	GEM forward	GEM gradient	DiGress forward	DeFoG forward
64	13.2 / 0.09	29.3 / 0.31	12.5 / 0.09	15.2 / 0.12
128	14.1 / 0.16	30.1 / 0.95	13.8 / 0.16	16.2 / 0.19
256	26.2 / 0.44	51.0 / 3.51	26.0 / 0.44	31.6 / 0.47

Table 10: **Runtime and memory scaling.** Time (ms) / peak memory (GiB) per network evaluation, on an NVIDIA H100 with batch size 1. GEM gradient includes the forward and backward passes.

F.4 Geodesic analysis: validity along geodesics

We compare GEM's energy-weighted geodesic to a cost-only baseline along interpolated paths between matched MOSES molecules. We align permutations using the node-matching c_{FGW} cost (Section 3, (3.4); node-matching details in Appendix C), and then use the aligned local cost c_{loc} (3.3) to evaluate distances, which is equivalent under the fixed alignment.

Geodesics. Paths are parameterized as cubic B-splines in the node/edge probability simplex with fixed endpoints and learned interior control points (8 control points, degree 3). Optimization minimizes the energy-weighted path length L_θ from Appendix H (with the $\exp(\beta V_\theta)$ weighting) plus an average-energy regularizer. We use $\beta = 0.1$, length weight $\lambda_L = 0.1$, energy weight $\lambda_{ME} = 1.0$, 2000 Adam iterations, and a learning rate of 10^{-3} [Kingma and Ba, 2015].

Validity is estimated by sampling 16 arc-length–uniform locations per path and drawing 16 discrete graphs per location. Validity is the fraction of RDKit-valid and connected samples, averaged along each path and then across pairs within each distance bin. We use three distance bins: $[5, 10)$, $[10, 15)$, and $[15, 20)$, with 256 molecules per bin.

F.5 Energy diagnostics

We evaluate a MOSES checkpoint trained with contrastive learning on independent random samples of 10,000 training and 10,000 held-out test molecules. Table 11 compares their energies with uniform categorical noise. Held-out molecules receive similar mean energies to training molecules, while noise has substantially higher energy.

Population	Energy (mean $\pm$ std.)
MOSES train	-274.26 ± 55.44
MOSES test	-273.88 ± 55.35
Uniform categorical noise	253.80 ± 58.12

Table 11: **Energy diagnostics.** Mean and standard deviation across samples under the same contrastively trained MOSES checkpoint.

We also run 1,024 data-initialized chains for 300 temperature-annealed mixing steps and save ten stages. Across these stages, mean energy and validity have correlation $r = -0.94$. Across sampler settings in a sweep using a contrastively trained checkpoint, terminal energy and V.U.N. have correlation $r = -0.72$. These correlations describe aggregate relationships across stages and sampler settings.

G Noise sensitivity of property regressors

Property regressor training. Let (x_1, ζ) denote a molecule (graph) and its scalar property (ζ). A clean regressor f_ϕ is trained on valid molecules by minimizing

$$\min_\phi \ \mathbb{E}_{(x_1,\zeta)\sim\mathcal{D}}\left[\|f_\phi(x_1) - \zeta\|_2^2\right]. \tag{G.1}$$

For noise-conditioned regressors, we sample a time $t \sim p(t)$ and a noisy graph $x_t \sim q_t(x_t \mid x_1)$ using the same discrete noising kernel as the forward process (marginal or uniform). The regressor takes (x_t, t) as input (time is fed as an additional scalar feature) and is trained with

$$\min_\phi \ \mathbb{E}_{(x_1,\zeta)\sim\mathcal{D},\ t\sim p(t),\ x_t\sim q_t(\cdot|x_1)}\left[\|f_\phi(x_t, t) - \zeta\|_2^2\right]. \tag{G.2}$$

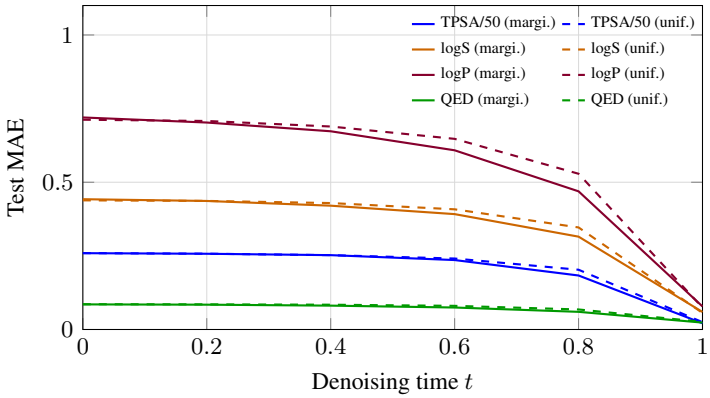

Figure 8: Test MAE of time/noise-conditioned property regressors versus denoising time t (0 = fully noised, 1 = clean/denoised). Each point averages 10k test molecules. Solid lines are marginal-noise regressors; dashed lines are uniform-noise regressors. TPSA MAE is scaled by $1/50$ for visualization. Errors are large for noisy inputs and only become reasonable around $t = 1$.

Regressor	logS $\geq$ -2.25 C.V.U.N. $\uparrow$	QED ≥ 0.9 C.V.U.N. $\uparrow$	logP ≤ 1.5 C.V.U.N. $\uparrow$
Clean	0.531	0.154	0.552
Noise-conditioned	**0.836**	**0.422**	**0.846**

Table 12: **Regressor ablations (GEM).** Clean regressors are time-independent and trained only on valid molecules. Noise-conditioned regressors are time-dependent and trained on uniformly noised graphs. We report C.V.U.N. for each property condition on MOSES.

During GEM sampling we do not explicitly track diffusion time, so we use an energy-based proxy to set t: we linearly increase t between the noise and data energy bands and clamp to $t = 1$ once the chain reaches the data distribution, using $t = 1$ for the remainder of generation.

Property regressors are trained on clean molecules, so evaluating them on noisy graphs is ill-posed when bonds and valences are corrupted. We measure the degradation by computing test mean absolute error (MAE) as a function of denoising time t along the diffusion path. We use 10k test molecules and report four representative properties (TPSA, logS, logP, QED). As expected, errors are large for noisy inputs and only become reasonable at $t = 1$ (fully denoised, clean samples). Figure 8 summarizes the trend. Solid lines are marginal-noise regressors; dashed lines are uniform-noise regressors.

H Geodesic analysis details

Path optimization. We represent a continuous path $\gamma(t)$ in the embedding space and optimize the energy-weighted length

$$L_\theta(\gamma) = \int_0^1 \exp\big(\beta V_\theta(\hat{\gamma}(t))\big) \|\dot{\hat{\gamma}}(t)\|_{\text{loc}}\, dt \tag{H.1}$$

with an average-energy regularizer $\lambda_{ME} \int_0^1 V_\theta(\hat{\gamma}(t))\, dt$. The cost-based baseline uses the same endpoint pairing and interpolates between the coupled graphs without the additional energy-weighting or average-energy regularization.

Spline parameterization. We implement geodesic paths using a cubic B-spline over learnable control-point logits for nodes and edges. The spline lives in the probability simplex (node/edge categorical distributions), and discrete graphs are sampled only for visualization and metric evaluation. The optimization minimizes the relative length ratio $L_{\text{spline}}/L_{\text{linear}} - 1$, where the segment lengths are weighted by $\exp\big(\beta V_\theta(x_\tau)\big)$ along the path.

Discrete evaluation. We discretize each continuous path at evenly spaced points in normalized arc length. At each point we sample categorical graphs from the path distribution to compute validity

and energy statistics. We compute average validity along each path and plot it against the endpoint distance induced by the FGW node-matching cost c_{FGW}. Validity estimation and binning details are in Appendix F.

I Contrastive loss objective derivation

Given the model density (where β_{mh} is a constant):

$$p_\theta(x) = \frac{e^{-\beta_{\mathrm{mh}} V_\theta(x)}}{Z_\theta}, \qquad Z_\theta = \sum_x e^{-\beta_{\mathrm{mh}} V_\theta(x)}. \tag{I.1}$$

let us consider maximum likelihood estimation:

$$\theta^\star = \arg\max_\theta \ \mathbb{E}_{x\sim p_{\text{data}}}\big[\log p_\theta(x)\big] \iff \theta^\star = \arg\min_\theta \ \mathcal{J}(\theta), \tag{I.2}$$

with

$$\mathcal{J}(\theta) := -\mathbb{E}_{p_{\text{data}}}\big[\log p_\theta(x)\big] = \mathbb{E}_{p_{\text{data}}}\big[\beta_{\mathrm{mh}} V_\theta(x)\big] + \log Z_\theta. \tag{I.3}$$

Taking gradients gives

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{p_{\text{data}}}\big[\beta_{\mathrm{mh}} \nabla_\theta V_\theta(x)\big] + \nabla_\theta \log Z_\theta, \tag{I.4}$$

where

$$\begin{aligned} \nabla_\theta \log Z_\theta = \frac{1}{Z_\theta}\nabla_\theta Z_\theta &= \frac{1}{Z_\theta}\sum_x e^{-\beta_{\mathrm{mh}} V_\theta(x)}\big(-\beta_{\mathrm{mh}}\nabla_\theta V_\theta(x)\big) \\ &= -\mathbb{E}_{p_\theta}\big[\beta_{\mathrm{mh}}\nabla_\theta V_\theta(x)\big]. \end{aligned} \tag{I.5}$$

Therefore, the gradient simplifies to

$$\boxed{\nabla_\theta \mathcal{J}(\theta) = \beta_{\mathrm{mh}}\left(\mathbb{E}_{p_{\text{data}}}\big[\nabla_\theta V_\theta(x)\big] - \mathbb{E}_{p_\theta}\big[\nabla_\theta V_\theta(x)\big]\right)}. \tag{I.6}$$

This motivates the contrastive loss objective $\mathcal{L}_{\mathrm{CL}}$ introduced in (3.15), where we approximate the expectation over p_θ using samples initialized either at uniform noise and data samples in 50%/50% proportions for general training, or exclusively at data samples (0%/100%) for the specialized training of data-initialized unconditional generation models. We then iteratively run a sequence of $(\text{greedy} + N_{\mathrm{CL}})$ proposals, explicitly detaching gradients flowing through the sampler transitions (i.e., Markov chain proposal and acceptance steps) to avoid backpropagation through the stochastic sampling process itself, ensuring stable estimation.

J Broader Impacts

GEM is a methodological contribution for generative modeling of discrete structured data, evaluated here on public graph benchmarks. Potential positive impacts include more efficient exploration of molecular design spaces, improved incorporation of structural or property constraints, and better tools for scientific discovery workflows. Potential negative impacts arise from the same generative capabilities: molecular generation methods could be misused to propose unsafe, toxic, or otherwise undesirable compounds if deployed without domain-specific screening and expert oversight. Our experiments do not validate generated molecules for synthesis, biological activity, safety, or deployability. Responsible use should therefore pair such methods with established cheminformatics filters, toxicity and synthesizability checks, domain-expert review, and application-specific governance before any real-world use.

K Additional molecular results

Table 13 compares additional MOSES baselines with GEM. We report these baselines separately from those in Table 1a due to differences in backbones and sampling methods. We recompute DiGress V.U.N. and FCD from released samples and derive Cometh's V.U.N. from reported validity, uniqueness, and novelty, using its predictor–corrector (PC) variant. Both baselines achieve lower FCD than our main GEM configuration, but with substantially lower V.U.N. (DiGress: 0.812 vs. 0.923).

Method	V.U.N. ↑	FCD ↓
DiGress [Vignac et al., 2023]	0.812	1.19
Cometh (PC) [Siraudin et al., 2025]	0.837	1.27
GEM	0.923	1.39

Table 13: **Additional molecular baselines on MOSES.** Noise-initialized generation. Higher V.U.N. and lower FCD are better.

Individual QM9 metrics. Table 14 reports individual QM9 metrics from the VFM and DeFoG papers and our GEM evaluation, with VFM's V.U.N. computed from its reported validity, uniqueness, and novelty. Higher novelty drives GEM's higher V.U.N., while VFM and DeFoG achieve lower FCD.

Method	Validity ↑	Uniqueness ↑	Novelty ↑	V.U.N. ↑	FCD ↓
VFM [Eijkelboom et al., 2024]	0.998	1.000	0.490	0.489	0.44
DeFoG [Qin et al., 2025]	0.993	0.963	0.364	0.348	0.81
GEM	0.987	0.991	0.626	0.612	0.92

Table 14: **Individual QM9 metrics.** Noise-initialized generation.

L Non-molecular graph generation

We evaluate GEM on Planar, comprising unlabeled 64-node graphs, and stochastic block model (SBM) graphs with up to 187 nodes. For Planar, we evaluate GEM and released DiGress samples using the native DiGress implementation for structural metrics and graph-isomorphism novelty [Vignac et al., 2023]. Both methods use 1,000 sampling steps. For SBM, we follow the DeFoG evaluation protocol and take baseline results from Table 7 of Qin et al. [2025]. Our SBM model uses the DeFoG backbone (17.5M parameters) with a permutation-invariant energy head (0.1M parameters), 1,000 sampling steps, and transport proposals with $N = 10$.

Method	Degree ↓	Clustering ↓	Orbit ↓	Spectral ↓	V.U.N. ↑
Planar (64 nodes)					
DiGress	0.000622	0.056330	0.009802	**0.006212**	**0.8500**
GEM	**0.000375**	**0.051806**	**0.005711**	0.007292	0.8075
SBM (up to 187 nodes)					
DiGress	0.0018	0.0485	0.0415	0.0045	0.600
DeFoG	$\mathbf{0.0006} \pm 0.0023$	0.0517 ± 0.0012	0.0556 ± 0.0739	0.0054 ± 0.0012	$\mathbf{0.900} \pm 0.051$
CatFlow (VFM)	0.0012	0.0498	0.0357	–	0.850
Cometh	0.0020 ± 0.0003	0.0498 ± 0.0000	0.0383 ± 0.0051	$\mathbf{0.0024} \pm 0.0003$	0.750 ± 0.037
GEM	0.0012 ± 0.0005	$\mathbf{0.0457} \pm 0.0008$	$\mathbf{0.0343} \pm 0.0033$	0.0048 ± 0.0005	$\mathbf{0.900} \pm 0.018$

Table 15: **Non-molecular graph generation.** The first four metrics are maximum mean discrepancies (MMD). SBM GEM results are means ± standard deviations over four seeds. Reported baseline uncertainties are retained, and a dash denotes an unavailable value. Best values within each benchmark are bold, including ties in reported means.

On Planar, GEM improves degree, clustering, and orbit MMD and validity (0.95 vs. 0.85), while DiGress achieves better spectral MMD, higher novelty (1.00 vs. 0.85), and higher V.U.N. scores. On SBM, GEM achieves the lowest mean clustering and orbit MMD among the listed methods and ties the best mean V.U.N. of 0.90. Overall, these results show that GEM remains competitive on these non-molecular graph benchmarks while providing an explicit interface to the learned unnormalized density landscape through its scalar energy.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: Our claims are restricted to the graph domain. The empirical evaluation focuses primarily on molecular datasets, with additional results on the non-molecular Planar and SBM datasets. We discuss the limitations and tradeoffs of the chosen representation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The discussion notes the main computational trade-off of energy-gradient sampling, while the appendix states the approximation and guarantee scope and reports robustness, sensitivity, and wall-clock measurements.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The paper states the assumptions and approximation scope behind the proposal derivations, and the appendix specifies that formal stationarity guarantees apply only to the MH mixing stage.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The paper describes datasets, metrics, algorithms, architecture, hyperparameters, sample counts, baseline handling, and robustness/runtime details; code is provided.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Code is provided to reproduce the paper's main claims.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Training schedules, optimizers, hyperparameters, sampling settings, architectures, sample counts, and evaluation protocols are specified in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The appendix reports mean and standard deviation for training-seed and sampling-seed variability, and states which sources of randomness are measured.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The appendix reports hardware for wall-clock experiments and provides training, sampling, and runtime details needed to estimate compute requirements.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses public molecular benchmarks and standard computational experiments, and does not involve human subjects or private data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Potential positive scientific uses and dual-use risks for molecular generation are discussed in Appendix J.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk pretrained models or scraped datasets requiring special safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: Existing datasets, software, and baselines are cited in the paper; released code will include the relevant license information.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The provided code documents how to reproduce the method.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method does not use LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.