EXISTENCE-FIELD DIFFUSION MODEL FOR SPATIAL POINT PROCESSES WITH VARIABLE CARDINALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

We study generative modeling of spatial point processes (SPP), where both the number of points and their spatial configuration are governed by a joint distribution. While diffusion models have achieved strong performance in modeling complex distributions, extending them to variable-cardinality SPP remains challenging. Existing approaches either sample cardinality before generating locations conditionally, or introduce specialized discrete operations to change the number of points during generation. We propose the existence-field diffusion model (EFDM), which associates each potential point with a continuous variable representing its degree of presence. EFDM uses Gaussian diffusion to jointly model point locations and cardinality through a simple continuous representation. Existence probabilities can increase or decrease in both forward and reverse diffusion, without explicit point-addition or point-deletion steps. The same construction naturally accommodates categorical attributes, allowing locations, existence, and attributes to be modeled together. Experiments demonstrate improved molecular stability and validity over the compared baselines on the molecular dataset, alongside competitive performance on real-world spatial and synthetic datasets.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021; Sohl-Dickstein et al., 2015) have achieved success in generative modeling, enabling high-quality generation in many domains. In applications such as image and audio generation, these models commonly operate on ordered arrays of a prescribed size. However, many real-world data are naturally represented as point sets with variable cardinality, such as spatial events, molecular structures, and particle systems. Modeling such data requires handling unordered point sets and learning distributions over both the number of points and their spatial configuration. Distributions over such objects are called spatial point processes (SPP). Diffusion models for SPPs pose a fundamental challenge: how to model both cardinality and spatial structure in a unified and permutation-invariant manner.

Real-world point sets exhibit dependencies among points as well as between cardinality and spatial configuration. In a molecule, bond lengths, bond angles, and steric constraints couple atomic positions: placing one atom restricts where others can lie, and all atoms must form a chemically consistent structure. Changing the number of atoms also changes the feasible molecular structures. In transportation data, point sets might represent e-scooter pick-up locations. Shared fluctuations in travel demand can produce spatial clusters, making observed locations informative about where other events are likely to occur and how many are expected. These examples motivate modeling the entire point set jointly, capturing both dependencies among points and their relationship with cardinality.

Point-process models capture event dependencies through conditional intensities (Hawkes, 1971; Mei and Eisner, 2017; Chen et al., 2021) or latent processes (Hong and Shelton, 2022). Diffusion models have also been applied to 3D shape generation with point clouds (Luo and Hu, 2021; Ji et al., 2024). In these tasks, point locations describe the shape's geometry, while the point count specifies its sampling resolution. For spatial events, molecules, and particle systems, cardinality is itself part of the random configuration to be modeled. To handle variable cardinality, a fixed-dimensional diffusion model (FDDM) (Hoogeboom et al., 2022) first samples a point count and then generates locations conditioned on that count. In particle physics, Mikuni et al. (2023) likewise first generate a particle count and properties of the collection as a whole, then conditionally generate the particles' features.

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

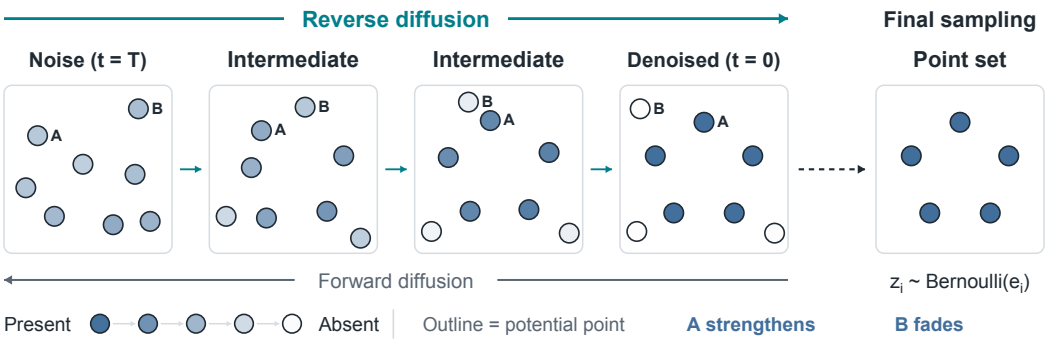


Figure 1: Binary EFDM jointly diffuses positions and existence. Outlines mark potential points; fill shading qualitatively indicates the degree of existence. In reverse diffusion, A strengthens and B fades. Final Bernoulli sampling yields the point set. Paths are schematic.

These hierarchical approaches can capture dependencies between cardinality and spatial configuration, but keep the sampled count fixed during point-level diffusion. In contrast, the trans-dimensional diffusion model (TDDM) (Campbell et al., 2023) and Point Set Diffusion (PSD) (Lüdke et al., 2025) jointly generate cardinality and locations, allowing both to evolve within a single generative process. However, they change point presence through discrete additions and deletions, rather than refining it gradually together with point locations.

We propose the existence-field diffusion model (EFDM), which brings variable-cardinality point sets into a Gaussian diffusion framework. Each potential point has a continuous existence variable that is diffused together with its coordinates (Fig. 1). A final readout converts the recovered fields into a discrete point set. Extending the existence field with categorical states also allows discrete attributes to be modeled in the same diffusion process. EFDM jointly refines point presence, geometry, and categorical attributes through Gaussian diffusion, without fixing cardinality before denoising as in FDDM or introducing explicit point additions and deletions as in TDDM and PSD.

On the molecular dataset QM9, EFDM jointly generates molecular geometry and categorical atom types with superior stability and validity relative to the reported FDDM and TDDM baselines. On real-world and synthetic datasets without point attributes, EFDM achieves competitive scores on joint count–configuration metrics. These comparisons provide empirical evidence of its ability to model such dependencies.

2 Related Work

Temporal and Spatial Point Processes. Existing point-process methods describe event distributions through intensity functions or latent processes. First-order intensity functions describe the expected local density of events but do not, by themselves, characterize interactions or determine the likelihood of an entire point configuration (Daley and Vere-Jones, 2003). In temporal point processes, history-dependent conditional intensities capture event interactions and determine event-sequence likelihoods; neural models parameterize them with recurrent networks (Mei and Eisner, 2017) or model event sequences with continuous-time normalizing flows (Chen et al., 2021). Other approaches introduce latent structure: Zhou et al. (2022) use latent dynamics to parameterize an intensity with closed-form integration, while Hong and Shelton (2022) build a hierarchy of latent Poisson processes.

Point Set Generation. For geometric data, existing point-set generators use a variety of mechanisms: PointFlow (Yang et al., 2019) uses continuous normalizing flows, PointGrow (Sun et al., 2020) generates points autoregressively, and ShapeGF (Cai et al., 2020) learns spatial gradient fields. In particular, PointFlow and ShapeGF represent shapes through distributions over spatial locations and can sample a user-selected number of points. This flexibility concerns the sampling resolution of a shape; these models do not learn the distribution of cardinality or its dependence on geometry.

Diffusion Models for Spatial Point Processes. For data with variable cardinality, some methods adopt a hierarchical generation process. The equivariant diffusion model (EDM) (Hoogeboom et al., 2022), referred to as the fixed-dimensional diffusion model (FDDM) in our comparisons, factorizes the point-set distribution as $p(x) = p(n)p(x \mid n)$: it first samples cardinality, then generates all points jointly using a cardinality-conditioned, permutation-equivariant denoiser. Mikuni et al. (2023) similarly use two diffusion models: one generates the particle count and global properties of the particle collection, such as momentum and mass, and the other jointly generates the particles' features conditioned on them. These hierarchical formulations can capture dependencies among points and between cardinality and spatial configuration, but the sampled cardinality remains fixed throughout point-level diffusion.

An alternative is to let cardinality and point locations evolve together during generation. Trans-dimensional diffusion (TDDM) (Campbell et al., 2023) combines continuous diffusion with discrete jumps. Its forward process diffuses coordinates and deletes points according to a prescribed rate, stopping deletions at a single point. The reverse process interleaves denoising with insertions governed by a learned, state-dependent jump rate and an insertion distribution. This design requires coordinating insertion with denoising so that newly added points have enough time to be refined. For molecules, the same process jointly diffuses coordinates and continuous encodings of atom types and charges.

Another approach, Point Set Diffusion (PSD) (Lüdke et al., 2025), evolves point sets through thinning and superposition. Its forward process independently removes data points and adds points from a noise process, eventually approaching that noise process. In reverse, a classifier identifies retained data points, while a conditional model predicts the number and locations of missing points, using a learned Gaussian mixture for locations on Euclidean domains. These predictions are combined with analytic thinning posteriors to remove noise points and restore data points. Retained coordinates remain fixed, so adjusting them requires deletion and resampling rather than gradual coordinate updates. The published experiments do not demonstrate joint generation of categorical point attributes.

Building on the goal of evolving cardinality and geometry together, our EFDM represents point presence continuously and denoises it jointly with coordinates and categorical attributes. This representation avoids the explicit point-insertion and point-deletion steps used by TDDM and PSD (Section 4.2).

3 Preliminaries

3.1 Spatial Point Processes

Let (D, d) be a complete separable metric space equipped with its Borel σ-algebra $\mathcal{B}$. A spatial point process (SPP) (Gelfand et al., 2010) is defined as a random configuration of points in D. Formally, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A spatial point process is a measurable mapping

$$x : \Omega \to \mathcal{N}_{\text{lf}}(D),$$

where $\mathcal{N}_{\text{lf}}(D)$ denotes the set of locally finite point configurations on D, i.e., subsets $x \subseteq D$ such that every bounded Borel set contains only finitely many points.

Throughout this work, we additionally assume finite total cardinality almost surely, $|x| < \infty$. A realization can then be represented as a finite point set:

$$x = \{x_1, \ldots, x_n\}, \quad x_i \in D,$$

where $n = |x|$ is the number of points.

3.2 Diffusion Models

Diffusion models are generative models that learn a data distribution by gradually diffusing a data point with random noise until it reaches a prior distribution. They then learn a reverse process that can reconstruct the original data distribution via denoising. In discrete time, the forward process is defined as a Markov chain $q(x^{(1:T)} \mid x^{(0)}) := \prod_{t=1}^{T} q(x^{(t)} \mid x^{(t-1)})$, where $q(x^{(t)} \mid x^{(t-1)})$ is the Markov diffusion kernel. The reverse process is defined as $p_\theta\big(x^{(0:T)}\big) := p\big(x^{(T)}\big) \prod_{t=1}^{T} p_\theta\big(x^{(t-1)} \mid x^{(t)}\big)$, approximated by a trained neural network with parameters θ.

4 EXISTENCE-FIELD DIFFUSION MODEL

We first formulate the problem of modeling variable-cardinality point sets, then introduce EFDM and its extension to categorical attributes.

4.1 PROBLEM FORMULATION

Specializing the setting in Section 3.1 to $D = \mathbb{R}^d$, let

$$\mathcal{X} = \{x \subset \mathbb{R}^d : |x| < \infty\}$$

denote the configuration space. Given a dataset $\mathcal{D}$ of i.i.d. samples from an unknown distribution $p^*(x)$ on $\mathcal{X}$, our goal is to learn a diffusion-based model $p_\theta(x)$ that approximates $p^*(x)$ and generates new point sets. The model must satisfy two requirements.

Permutation invariance. When operating on ordered point representations, the model must assign the same density to every ordering of a configuration. For any permutation π of $\{1, \ldots, n\}$, this requires $p_\theta(x_1, \ldots, x_n) = p_\theta(x_{\pi(1)}, \ldots, x_{\pi(n)})$.

Variable cardinality. The model must learn the joint distribution of cardinality and spatial configuration, capturing dependencies among points as well as between their arrangement and point count.

4.2 EXISTENCE-FIELD POINT DIFFUSION

We introduce the existence-field diffusion model (EFDM) for variable-cardinality point sets. EFDM jointly diffuses point coordinates and continuous existence variables, allowing spatial structure and cardinality to be generated together without specifying the point count in advance or relying on discrete point additions and deletions during diffusion. We first describe binary existence, then extend the representation to points with categorical attributes.

4.2.1 BINARY EXISTENCE FIELDS

For each potential point $x_i \in \mathbb{R}^d$, we introduce a real-valued existence variable $u_i \in \mathbb{R}$. Its existence probability is $e_i = \sigma(u_i) \in (0, 1)$, where σ is the sigmoid function. The joint diffusion state is

$$y = (y_1, \ldots, y_N) \in (\mathbb{R}^{d+1})^N, \qquad y_i = (x_i, u_i). \tag{1}$$

We represent each configuration using a fixed budget of N potential points, so EFDM models distributions on the bounded-cardinality space

$$\mathcal{X}_N = \{x \subset \mathbb{R}^d : |x| \le N\} \subset \mathcal{X}. \tag{2}$$

In principle, N can be chosen arbitrarily large. In practice, we choose a finite slot budget to control the size of the diffusion state and meet computational and memory budgets. In the experiments, N is chosen to be at least as large as the maximum cardinality in the training data, so that no observed point set is truncated.

To represent a point set $x = \{x_1, \ldots, x_n\}$ with $n \le N$, we assign its points to n active slots and use the remaining slots as auxiliary padding. Active and inactive slots receive positive and negative existence logits, respectively. The encoding and padding settings used in the experiments are specified in Appendix A.2.

Joint diffusion. In the discrete-time formulation, we use the Gaussian forward process of a denoising diffusion probabilistic model (DDPM) (Ho et al., 2020) together with an existence-weighted denoising objective:

$$y^{(t)} = \sqrt{\bar{\alpha}_t}\, y^{(0)} + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{3}$$

$$\mathcal{L}(\theta) = \mathbb{E}_{y^{(0)}, t, \epsilon}\left[\frac{1}{Nd}\sum_{i=1}^{N} \sigma(u_i^{(t)})\, \|\epsilon_{x,i} - \hat{\epsilon}_{x,i}\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} (\epsilon_{u,i} - \hat{\epsilon}_{u,i})^2\right], \tag{4}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, $\epsilon_i = (\epsilon_{x,i}, \epsilon_{u,i})$, and $(\hat{\epsilon}_{x,i}, \hat{\epsilon}_{u,i}) = \epsilon_\theta(y^{(t)}, t)_i$. The coordinate denoising error is weighted by the existence probability of each noisy slot, while the coordinate and existence losses are averaged separately. Here $y^{(0)}$ is sampled from the encoded training data distribution, t is uniform on $\{1, \dots, T\}$, and the noise schedule $\beta_s \in (0, 1)$ is chosen so that the terminal forward distribution is approximately standard Gaussian. In this discrete-time formulation, sampling starts from $y^{(T)} \sim \mathcal{N}(0, I)$ and uses DDPM reverse transitions parameterized by ϵ_θ. The representation also supports continuous-time variance-preserving (VP) diffusion (Song et al., 2021) within the same Gaussian diffusion framework. The formulation and sampler used for each dataset are specified in Appendix A.2. A single denoiser processes the full state y, jointly modeling spatial structure and existence. We require slot-permutation equivariance, $\epsilon_\theta(Py, t) = P\epsilon_\theta(y, t)$ for any slot permutation P. Together with the exchangeable Gaussian prior and identical decoding rules for all slots, this yields a permutation-invariant point-set distribution.

After denoising, write the generated state as $\hat{y}_i^{(0)} = (\hat{x}_i, \hat{u}_i)$. Bernoulli readout gives

$$\hat{z}_i \sim \mathrm{Bernoulli}(\sigma(\hat{u}_i)), \qquad \hat{x} = \{\hat{x}_i : \hat{z}_i = 1\}, \qquad \hat{n} = \sum_{i=1}^{N} \hat{z}_i \le N. \tag{5}$$

4.2.2 Extension to Attributes

For a marked point set, each active point carries $M \ge 1$ categorical attributes. Attribute $m \in \{1, \dots, M\}$ has K_m meaningful values, with $a_i^{(m)} \in \{1, \dots, K_m\}$ denoting its value for point i. We retain one existence variable u_i per slot, with $e_i = \sigma(u_i)$, and let $\boldsymbol{\rho}_i^{(m)}$ describe the conditional distribution of attribute m given that the point exists. On the simplex interior, $\rho_{ik}^{(m)} > 0$ and $\sum_{k=1}^{K_m} \rho_{ik}^{(m)} = 1$. Introducing category 0 for non-existence gives the full categorical field

$$\gamma_{i0}^{(m)} = 1 - e_i, \qquad \gamma_{ik}^{(m)} = e_i \rho_{ik}^{(m)}, \quad k = 1, \dots, K_m. \tag{6}$$

Thus $\boldsymbol{\gamma}_i^{(m)} = (\gamma_{i0}^{(m)}, \dots, \gamma_{iK_m}^{(m)})$ lies in the probability simplex over $K_m + 1$ categories. The same scalar existence probability e_i is shared across all attributes of slot i, so the slot is jointly present or absent for all attributes.

To obtain unconstrained attribute coordinates, we fix category 1 as the reference within each conditional distribution:

$$v_{i1}^{(m)} = 0, \qquad v_{ik}^{(m)} = \log \frac{\rho_{ik}^{(m)}}{\rho_{i1}^{(m)}}, \quad k = 2, \dots, K_m. \tag{7}$$

The free coordinates are $\boldsymbol{v}_i^{(m)} = (v_{i2}^{(m)}, \dots, v_{iK_m}^{(m)}) \in \mathbb{R}^{K_m - 1}$, which uniquely determine $\boldsymbol{\rho}_i^{(m)} = \mathrm{softmax}(0, \boldsymbol{v}_i^{(m)})$. The joint diffusion state is

$$y_i = (x_i, u_i, \boldsymbol{v}_i^{(1)}, \dots, \boldsymbol{v}_i^{(M)}) \in \mathbb{R}^{d+1+\sum_{m=1}^{M}(K_m - 1)}. \tag{8}$$

We apply the Gaussian forward process in Eq. (3) to this state and extend the denoising objective to include errors on the free attribute coordinates. A single permutation-equivariant denoiser processes all slots, jointly modeling coordinates, existence, and attributes. With no attributes, the state reduces to binary EFDM. The experimental encoding of categorical labels is described in Appendix A.2.

After denoising, we draw $\hat{z}_i$ once per slot as in Eq. (5). For each active slot $\hat{z}_i = 1$, we draw $\hat{a}_i^{(m)} \sim \mathrm{Categorical}(\hat{\boldsymbol{\rho}}_i^{(m)})$ and return $\hat{x} = \{(\hat{x}_i, \hat{a}_i^{(1)}, \dots, \hat{a}_i^{(M)}) : \hat{z}_i = 1\}$. Inactive slots and all their attributes are discarded. The categorical draws are independent conditional on the generated fields, while the joint denoiser can model dependencies among the fields. Alternatively, the generated categorical fields can be decoded deterministically; the parameterization and readout used for QM9 are specified in Section 5.3.1.

5 Experiments

We evaluate EFDM on molecular generation and spatial point-set modeling. The main experiments cover the QM9 molecule dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014), the

Japan Earthquakes dataset (U.S. Geological Survey, 2024), and the City of Louisville e-scooter trip dataset (Louisville Metro Government, 2019). We also use a controlled bent Gaussian-mixture synthetic benchmark to study count–configuration dependence (Section 5.3.4).

5.1 Baselines

We consider five baselines: FDDM, TDDM, PSD, Independent Point Diffusion (IDM), and cardinality-adaptive TDDM (TDDMv). FDDM, TDDM, and PSD are described in Section 2.

IDM samples cardinality separately and models points independently via $p(x \mid n) \approx \prod_{i=1}^{n} p_{\mathrm{pt}}(x_i)$, where p_{pt} is a pointwise diffusion model shared across cardinalities.

Following the sample-independent rate design of Campbell et al. (2023), our base TDDM uses a shared deletion rate calibrated using the maximum training cardinality minus one. Forward deletion counts are capped to retain at least one point. QM9 contains at most 29 atoms per molecule, including hydrogens (Hoogeboom et al., 2022). To better accommodate the larger point sets in the Earthquake and Trip datasets, we introduce TDDMv, a variant that adapts the deletion rate to each sample's initial cardinality. It uses the observed initial count during training and the model's current prediction of that initial count during generation. Dataset-specific configurations are provided in Appendix A.2.

5.2 Evaluation Metrics

For QM9, we report atom stability, molecule stability, and validity (Section 5.3.1). For spatial point sets, we assess cardinality, spatial statistics, and their combined distributional behavior. Count W_1 measures cardinality accuracy by comparing the distributions of the number of points per set. Mean, Var, Skew, Kurt, and NN1 assess spatial statistics: location, spread, asymmetry, excess kurtosis, and local spacing, respectively. We report W_1 distances between the generated and held-out distributions of these per-set summaries, with coordinate-wise discrepancies averaged over dimensions. MMD, JMMD^2, and ICMMD^2 complement these separate summaries with distribution-level comparisons: MMD compares whole point sets, while JMMD^2 and ICMMD^2 explicitly relate cardinality to spatial configuration through joint and conditional comparisons, respectively.

We adopt maximum mean discrepancy (MMD) from Gretton et al. (2012) and reuse the implementation in PSD (Lüdke et al., 2025), which constructs a point-set kernel using Wasserstein distance (WD), implemented with 50-projection sliced W_2. PSD estimates its bandwidth from pairwise distances involving real test and generated sets. To complement this evaluation-dependent choice, we report $\mathrm{MMD}_{\text{train-scale}}$, whose kernel scale is fixed from training data and shared across methods; its scale convention is specified in Appendix A.4.

However, this Wasserstein-based MMD uses normalized point measures without an explicit count term and does not separately assess spatial agreement at each cardinality. We therefore use JMMD^2 (Long et al., 2017) to compare the joint distribution of cardinality and spatial configuration using an exact-count product kernel. ICMMD^2, adapted from Yan et al. (2024), compares spatial distributions conditional on exact cardinality, weighted by eligible real count frequencies. It additionally penalizes insufficient generated support. These two metrics report squared scores. Full definitions, preprocessing, bandwidths, estimators, and coverage diagnostics are given in Appendix A.4; dataset-specific evaluation diagnostics are given in Appendix A.5.

The Earthquake and Trip tables report the same ten metrics: Mean, Var, Skew, Kurt, NN1, Count W_1, MMD, train-scale MMD, JMMD^2, and ICMMD^2. The bent synthetic table in Appendix A.1 reports the same ten metrics. We retain unnormalized Count W_1 throughout. JMMD^2 is displayed after multiplication by 10^3. Across result tables, bold marks the best reported value (the mean for repeated evaluations), including ties at the displayed precision. Rankings are descriptive and do not indicate statistical significance. We do not rank the near-saturated Trip MMD.

5.3 Experimental Results

5.3.1 Molecular Generation on QM9

We evaluate unconditional molecular generation on QM9, which contains small organic molecules with varying numbers of atoms. Each molecule is represented as a set of atoms with 3D coordinates

Table 1: Unconditional molecular generation on QM9. All metrics are percentages (↑), evaluated on 10,000 generated molecules. FDDM and TDDM scores are taken from Table 2 of Campbell et al. (2023); the FDDM scores originate from Hoogeboom et al. (2022). EFDM is evaluated once using the final-epoch EMA checkpoint. Bold marks the highest reported value at the displayed precision, including ties.

Method	Atom stability	Molecule stability	Validity
EFDM	**99.04**	**91.18**	**96.38**
FDDM	98.7	82.0	91.9
TDDM	98.3	87.2	92.3

and categorical atom types. For QM9, we jointly diffuse atom coordinates and direct categorical logits relative to a fixed zero logit for the inactive category. Training uses $+1$ for the observed atom type and -1 for all other type logits, with all type logits -1 for inactive slots. Since atom type is the only attribute, a deterministic argmax over the type and inactive logits jointly determines presence and type without additional sampling; Appendix A.3 details the parameterization and readout. We use continuous-time VP diffusion with predictor–corrector sampling and an E(n)-equivariant graph neural network (EGNN) backbone (Satorras et al., 2021) together with a Transformer component (Vaswani et al., 2017); implementation details are given in Appendix A.2.

We compare EFDM with the published FDDM and TDDM scores from Table 2 of Campbell et al. (2023). The FDDM baseline corresponds to the equivariant diffusion model (EDM) of Hoogeboom et al. (2022). We report the same three molecular quality metrics: atom stability, molecule stability, and validity. Bonds are inferred from interatomic distances. Atom stability measures the fraction of atoms with the correct valency, and molecule stability measures the fraction of molecules for which all atoms are stable; validity is assessed using RDKit.

Using the final exponential moving average (EMA) checkpoint at epoch 8,000, EFDM achieves 99.04% atom stability, 91.18% molecule stability, and 96.38% validity, exceeding the reported FDDM and TDDM scores on all three metrics (Table 1). Compared with TDDM, molecule stability and validity improve by 4.0 and 4.1 percentage points, respectively.

5.3.2 Earthquake Data

We use the Japan Earthquakes dataset from the earthquake catalog maintained by the U.S. Geological Survey (2024), following the dataset used by Point Set Diffusion (PSD) (Lüdke et al., 2025). Each observation is a spatial point set of earthquake locations within a week. The 1,500 weekly sets are randomly divided into 900 training, 300 validation, and 300 test sets for each of three split/training seeds. This evaluates generation of weekly spatial patterns rather than forecasting future weeks.

We compare EFDM with FDDM, TDDM, TDDMv, IDM, and PSD. FDDM and IDM sample cardinalities from a distribution fitted to the training counts before coordinate generation; EFDM and the trans-dimensional baselines determine cardinality within the generative process. All methods are trained for 60,000 epochs, and each checkpoint is selected by the lowest EMA validation loss. We sample without a corrector, rejecting empty proposals and resampling until 4,096 nonempty point sets are obtained per method and seed. All ten metrics, including count W_1, use these accepted sets against all 300 test sets. Implementation and evaluation details are given in Appendix A.2.

Table 2 shows that EFDM achieves lower mean errors than PSD across all ten reported metrics and strong overall performance among the evaluated methods. EFDM obtains the best mean scores for train-scale MMD, ICMMD, and three of the five spatial-summary metrics (Mean, Skewness, and Kurtosis). EFDM is consistently second-best in variance, NN1, MMD, and JMMD, with FDDM leading these metrics through a two-stage formulation that separately fits the marginal cardinality distribution $p(n)$ before conditional coordinate generation. For Count W_1, FDDM and IDM use the same separately fitted cardinality model to sample n before coordinate generation, whereas EFDM generates cardinality and spatial structure jointly within a single diffusion process. Despite not using a separate cardinality model, EFDM achieves a Count W_1 of 6.2 and a JMMD^2 of 0.407×10^{-3}. These results support continuous existence fields as an effective approach to joint cardinality–spatial modeling.

Table 2: Japan Earthquakes. All ten metrics use the same 4,096 accepted nonempty generated sets per run against 300 test sets. Values are mean ± sample standard deviation across three split/training seeds, computed directly from the unrounded per-seed metrics. All errors are lower-is-better; bold marks the lowest mean at the displayed precision, including ties.

Method	Mean	Var	Skew	Kurt	NN1
EFDM	$\mathbf{0.0271 \pm 0.0128}$	0.0295 ± 0.0116	$\mathbf{0.0547 \pm 0.0130}$	$\mathbf{0.1869 \pm 0.0110}$	0.0097 ± 0.0022
FDDM	0.0290 ± 0.0155	$\mathbf{0.0291 \pm 0.0094}$	0.0582 ± 0.0143	0.1910 ± 0.0369	$\mathbf{0.0045 \pm 0.0011}$
TDDM	0.1993 ± 0.0136	0.8057 ± 0.0177	1.5626 ± 0.2349	13.3865 ± 0.4090	0.0932 ± 0.0046
TDDMv	0.0441 ± 0.0174	0.0374 ± 0.0080	0.1374 ± 0.0217	0.4117 ± 0.0653	0.0354 ± 0.0026
IDM	0.1044 ± 0.0155	0.0827 ± 0.0069	0.2348 ± 0.0165	0.6124 ± 0.0631	0.0212 ± 0.0035
PSD	0.0563 ± 0.0119	0.0450 ± 0.0094	0.1305 ± 0.0136	0.3537 ± 0.0608	0.0452 ± 0.0068

Method	Count W_1	MMD	$\mathrm{MMD}_{\text{train-scale}}$	JMMD^2 ($\times 10^3$)	ICMMD^2
EFDM	6.2 ± 1.5	0.0701 ± 0.0118	$\mathbf{0.0498 \pm 0.0142}$	0.407 ± 0.197	$\mathbf{0.0085 \pm 0.0105}$
FDDM	$\mathbf{4.1 \pm 0.3}$	$\mathbf{0.0679 \pm 0.0105}$	0.0500 ± 0.0135	$\mathbf{0.265 \pm 0.224}$	0.0103 ± 0.0069
TDDM	224.5 ± 20.2	0.0729 ± 0.0058	0.8078 ± 0.0105	23.821 ± 4.018	1.5419 ± 0.0271
TDDMv	27.9 ± 2.9	0.0798 ± 0.0120	0.0751 ± 0.0176	5.732 ± 0.726	0.3130 ± 0.0149
IDM	4.7 ± 0.6	0.1249 ± 0.0052	0.1752 ± 0.0133	0.694 ± 0.184	0.0579 ± 0.0111
PSD	18.3 ± 5.7	0.0922 ± 0.0095	0.0924 ± 0.0111	1.890 ± 0.360	0.0700 ± 0.0288

5.3.3 TRIP DATA

We use the City of Louisville e-scooter trip dataset, which contains real-world shared micro-mobility trip records in Louisville, Kentucky. Each sample consists of approximately 10% of the pick-up locations for a given day, after filtering using statistics fitted on the training data. For each split/training seed, the days are randomly divided into 268 training, 89 validation, and 91 test sets.

We compare EFDM with FDDM, TDDM, TDDMv, IDM, and PSD, with the method-specific configurations detailed in Appendix A.2. All methods are trained for 60,000 epochs and evaluated at the checkpoint with the lowest EMA validation loss. We sample without a corrector, rejecting empty proposals and resampling until 4,096 nonempty point sets are obtained per method and seed. All ten metrics, including count W_1, use these accepted sets against the full 91-set test split. We report mean ± sample standard deviation over the three seeds.

EFDM achieves the lowest mean train-scale MMD, JMMD^2, and ICMMD^2 scores among the compared methods (Table 3). It also achieves the lowest mean errors on four of the five spatial-summary metrics and ranks second on kurtosis.

Evaluation details are in Appendix A.5. Coordinate normalizations and test-set sizes differ, so absolute errors are compared only within each dataset.

5.3.4 SYNTHETIC DATA: BENT GAUSSIAN MIXTURES

In this two-component Gaussian benchmark, a latent variable couples cardinality and component separation. The fixed pre-bend width is $\sigma_{\text{data}} = 0.2$; the invertible bend $(x, y) \mapsto (x, y + x^2)$ preserves their dependence.

Figure 2 compares count and estimated separation after the same inverse bend. At larger counts, EFDM tracks the real-test trend more closely than FDDM and PSD, whose curves flatten; IDM is nearly count-independent, TDDM covers a narrow count range, and TDDMv favors lower counts.

EFDM achieves the lowest JMMD^2 and full ICMMD^2 scores, while FDDM and PSD lead on several spatial summaries (Table 4). Full data and evaluation details are provided in Appendix A.1.

Table 3: Louisville Trip: all metrics use the same 4,096 accepted nonempty generated sets and 91 test sets. Values are mean ± sample standard deviation over three seeds; lower is better. Bold marks the lowest displayed mean, including ties. The original MMD is nearly constant across methods and is therefore not ranked. The unbiased JMMD^2 estimate may be negative.

Method	Mean	Var	Skew	Kurt	NN1
EFDM	$\mathbf{0.0576 \pm 0.0090}$	$\mathbf{0.0602 \pm 0.0103}$	$\mathbf{0.1138 \pm 0.0437}$	0.3805 ± 0.0201	$\mathbf{0.0151 \pm 0.0119}$
FDDM	0.0635 ± 0.0129	0.0842 ± 0.0306	0.1150 ± 0.0490	$\mathbf{0.3697 \pm 0.0663}$	0.0159 ± 0.0083
TDDM	0.3348 ± 0.0055	0.2323 ± 0.0339	0.5648 ± 0.0661	1.9065 ± 0.1142	0.0632 ± 0.0104
TDDMv	0.1024 ± 0.0254	0.1050 ± 0.0088	0.2062 ± 0.0416	0.6367 ± 0.2265	0.0456 ± 0.0070
IDM	0.2042 ± 0.0132	0.1467 ± 0.0103	0.3057 ± 0.0279	0.7087 ± 0.1727	0.0304 ± 0.0085
PSD	0.0647 ± 0.0104	0.0787 ± 0.0161	0.1295 ± 0.0307	0.3976 ± 0.1282	0.0343 ± 0.0082

Method	Count W_1	MMD	$\text{MMD}_{\text{train-scale}}$	JMMD^2 $(\times 10^3)$	ICMMD^2
EFDM	13.81 ± 7.36	$0.106005 \pm 2.04 \times 10^{-6}$	$\mathbf{0.0756 \pm 0.0033}$	$\mathbf{-0.151 \pm 1.452}$	$\mathbf{0.0123 \pm 0.0085}$
FDDM	$\mathbf{10.93 \pm 2.92}$	$0.105997 \pm 4.67 \times 10^{-6}$	0.0782 ± 0.0041	0.145 ± 1.515	0.0195 ± 0.0236
TDDM	141.99 ± 10.55	$0.105987 \pm 1.11 \times 10^{-11}$	0.4050 ± 0.0149	23.296 ± 1.319	1.5403 ± 0.0807
TDDMv	50.81 ± 8.12	$0.106130 \pm 5.25 \times 10^{-5}$	0.1292 ± 0.0126	5.921 ± 2.890	0.2229 ± 0.2112
IDM	$\mathbf{10.93 \pm 2.92}$	$0.105987 \pm 1.83 \times 10^{-7}$	0.2427 ± 0.0064	0.802 ± 1.619	0.1191 ± 0.0172
PSD	$\mathbf{10.93 \pm 6.14}$	$0.106001 \pm 6.14 \times 10^{-6}$	0.0866 ± 0.0049	0.240 ± 1.718	0.0126 ± 0.0192

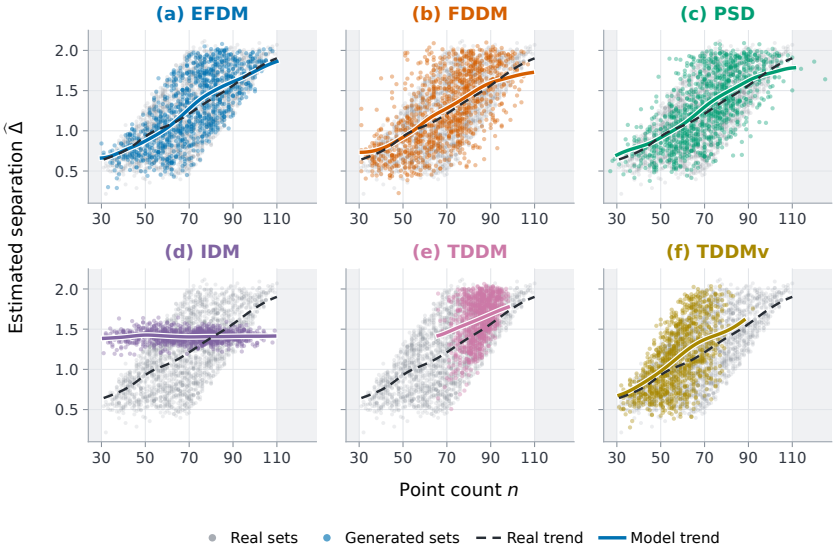


Figure 2: Bent synthetic count versus estimated separation. Each panel compares 1,024 generated sets (colored) with the same 2,000 real test sets (gray); solid/dashed curves show their trends. Shading marks counts outside $[30, 110]$. One training seed; estimator and smoothing details are in Appendix A.1.

6 Conclusions

EFDM brings variable-cardinality point sets into a unified Gaussian diffusion framework, allowing point presence, geometry, and categorical attributes to be refined jointly without fixing cardinality before denoising or introducing explicit birth–death transitions. Its improved stability and validity on the molecular dataset and competitive performance on real-world and synthetic spatial datasets demonstrate the potential of continuous existence fields for point-set generation. In practice, the chosen slot budget sets the maximum supported cardinality, and larger budgets involve additional computation and memory.

AI USE STATEMENT

AI tools were only used to assist with manuscript editing and polishing, code implementation and debugging for experiments and synthetic data generation, and the presentation of mathematical derivations. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

Details required to reproduce the experiments are provided in the supplementary material. Appendix A.1 specifies the synthetic-data construction and evaluation protocol; Appendices A.2–A.3 describe model configurations, training, sampling, and categorical parameterizations; and Appendices A.4–A.5 provide the complete metric definitions and spatial evaluation protocols. Anonymous source code for training, sampling, and evaluation is provided in the supplementary material.

ETHICS STATEMENT

This work develops generative modeling methodology using synthetic and publicly available scientific and spatial datasets. It does not involve human-subject experiments or collection of private personal data.

REFERENCES

Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *European Conference on Computer Vision*, pages 364–381. Springer, 2020.

Andrew Campbell, William Harvey, Christian Weilbach, Valentin De Bortoli, Thomas Rainforth, and Arnaud Doucet. Trans-dimensional generative modeling via jump diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pages 42217–42257, 2023.

Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XQQA6-So14.

Daryl J. Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer, 2003.

Alan E. Gelfand, Peter Diggle, Peter Guttorp, and Montserrat Fuentes, editors. *Handbook of Spatial Statistics*. CRC Press, 2010.

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

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. doi: 10.1093/biomet/58.1.83.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

Chengkuan Hong and Christian Shelton. Deep Neyman-Scott processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3627–3646. PMLR, 2022.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

Junzhong Ji, Runfeng Zhao, and Minglong Lei. Latent diffusion transformer for point cloud generation. *The Visual Computer*, 40(6):3903–3917, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2208–2217, 2017. URL https://proceedings.mlr.press/v70/long17a.html.

Louisville Metro Government. City of Louisville e-scooter trip data. https://www.kaggle.com/datasets/busielmorley/city-lousiville-escooter-trip-data, 2019.

David Lüdke, Enric Rabasseda Raventós, Marcel Kollovieh, and Stephan Günnemann. Unlocking point processes through point set diffusion. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4anfpHj0wf.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3D point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021.

Hongyuan Mei and Jason M. Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Vinicius Mikuni, Benjamin Nachman, and Mariel Pettee. Fast point cloud generation with diffusion models in high energy physics. *Physical Review D*, 108(3):036025, 2023.

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

RDKit. RDKit: Open-source cheminformatics. https://www.rdkit.org. Accessed: 2026-09-22.

Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9323–9332. PMLR, 2021. URL https://proceedings.mlr.press/v139/satorras21a.html.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. PointGrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–70, 2020.

U.S. Geological Survey. Search earthquake catalog. https://earthquake.usgs.gov/earthquakes/search/, 2024. Website accessed: 2026-09-23.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Jian Yan, Zhuoxi Li, and Xianyang Zhang. Distance and kernel-based measures for global and local two-sample conditional distribution testing. arXiv preprint arXiv:2210.08149v2. https://arxiv.org/abs/2210.08149v2, 2024.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019.

Zihao Zhou, Xingyi Yang, Ryan Rossi, Handong Zhao, and Rose Yu. Neural point process for learning spatiotemporal event dynamics. In *Learning for Dynamics and Control Conference*, pages 777–789. PMLR, 2022.

A Supplementary Material

This appendix provides bent synthetic data generation details and complete experimental results, model configurations, QM9 evaluation, and spatial evaluation metrics and protocols, together with supplementary visualizations for the QM9, Earthquake, and Trip experiments.

A.1 Bent Gaussian Mixtures: Data and Results

This section gives the data-generation rules, evaluation protocol, and complete results for the bent Gaussian-mixture benchmark introduced in Section 5.3.4, together with estimation and smoothing details for Figure 2.

Base distribution. Each point set has two independent latent variables $a, b \overset{\text{iid}}{\sim} \mathrm{Unif}(0,1)$. They determine its cardinality and component separation as

$$n = \mathrm{round}(30 + 40(a+b)), \qquad \Delta = 0.5 + 1.5a, \qquad \sigma_{\mathrm{data}} = 0.2. \tag{9}$$

Thus $30 \le n \le 110$, and cardinality is statistically related to separation without determining it uniquely. Conditional on these latents, the unbent points are sampled independently as

$$\boldsymbol{\xi}_i \mid a, b \overset{\text{iid}}{\sim} \tfrac{1}{2}\mathcal{N}\left(\begin{pmatrix}-\Delta/2\\0\end{pmatrix}, \sigma_{\mathrm{data}}^2 I_2\right) + \tfrac{1}{2}\mathcal{N}\left(\begin{pmatrix}\Delta/2\\0\end{pmatrix}, \sigma_{\mathrm{data}}^2 I_2\right), \qquad i = 1, \dots, n. \tag{10}$$

Component assignments are independent Bernoulli draws: the two components have equal sampling probability, but are not forced to contain equal numbers of points in an individual set. We randomly permute the sampled points before storing the set.

Nonlinear deformation and pairing. The observed points are obtained through the pointwise shear

$$\boldsymbol{x}_i = T_\kappa(\boldsymbol{\xi}_i), \qquad T_\kappa(r,s) = (r, s + \kappa r^2), \qquad \kappa = 1. \tag{11}$$

Its inverse is $T_\kappa^{-1}(x,y) = (x, y - \kappa x^2)$, and its Jacobian determinant is one. The transformation changes the geometry while retaining the information in the original point set and its cardinality. It does not introduce additional interactions between points: conditional independence given the shared latents is preserved. The fixed σ_{data} describes the Gaussian width before bending, not a constant local Euclidean width after the nonlinear transformation.

The bent and unbent datasets are paired point by point. They retain the same latent draws, cardinalities, component labels, point order, and train/validation/test membership. We use 500 training, 200 validation, and 2,000 test sets. The latent variables and generating parameters are retained as diagnostic metadata, not additional model inputs. The transformation is evaluated in float64 before converting the coordinates to float32; existing $+\infty$ padding is retained and is not transformed. No centering, rescaling, or clipping is applied to construct the bent dataset. The bounds used by the evaluation adapter are recomputed from the bent training split.

Training and evaluation. All six methods use the same 500 training, 200 validation, and 2,000 test sets. The six reported models are trained from scratch for 60,000 epochs with Adam (Kingma and Ba, 2015) and training seed 2025. The set-based models use four-layer networks with 128-dimensional embeddings, 256-dimensional feed-forward widths, and four attention heads; IDM uses a four-layer point-wise MLP without attention. Training batches contain 64 point sets for EFDM, FDDM, TDDM, TDDMv, and PSD, whereas IDM uses 4,096 individual points per batch. The learning rate is 10^{-4} with zero weight decay except for PSD, which uses learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 2. EMA decays by 0.999 after each update. Every 100 epochs, we evaluate the EMA model with fixed validation seed 123 and select the lowest-loss checkpoint among 600 candidates for each method. Each selected model generates 1,024 point sets with sampling seed 42, which are compared against all 2,000 test sets. Non-PSD models use 1,000 predictor steps without a corrector; PSD uses its native 100-step sampler. These are single-run results, not means over training seeds. All 1,024 generated sets for each method are nonempty.

Feature W_1 is evaluated in the observed bent coordinates, using all generated sets, including any sets with cardinality outside $[30, 110]$. Cardinality W_1 uses every generated set. Structural spacing and width diagnostics first apply the same inverse map $T_1^{-1}(x, y) = (x, y - x^2)$ to both real and generated points. For JMMD, ICMMD, and the spatial MMD variants, the evaluation adapter maps coordinates to a bounded domain and clips values outside its training-derived bounds. This metric preprocessing is distinct from the unclipped data-generation process.

JMMD and ICMMD use the shared training-derived baseline bandwidth $\sigma = 0.133825$ and the definition in Appendix A.4. Train-scale MMD uses $h = 0.161057$, the median distance over all ordered training pairs including self-pairs, with kernel $\exp[-d_{\mathrm{SW}}/(2h)]$. Here d_{SW} denotes the unsquared sliced-Wasserstein distance. The bandwidths are shared across methods; none is fitted to a method's test performance.

Quantitative results. Table 4 shows that EFDM achieves the lowest JMMD^2 and full ICMMD^2 scores. Its measured joint discrepancy is 68.3% lower than FDDM's and 55.9% lower than PSD's. FDDM has the lowest cardinality error and both MMD variants, as well as the lowest mean, variance, and kurtosis summary errors; PSD has the lowest skewness and NN1 errors. These results illustrate that strong count and spatial-summary scores can coexist with a larger discrepancy under the joint metric. EFDM retains the lowest JMMD^2 at half and twice the training bandwidth (Table 5).

Table 4: Bent Gaussian mixtures with fixed pre-transform $\sigma_{\mathrm{data}} = 0.2$ and $\kappa = 1$. Results use a single training seed and 1,024 generated sets against 2,000 test sets. The upper panel reports spatial-summary W_1 in the observed bent coordinates. All errors are lower-is-better. Bold marks the lowest value at the displayed precision, including ties.

Method	Mean	Var	Skew	Kurt	NN1
EFDM	0.02250	0.02214	0.02432	0.04479	0.001179
FDDM	**0.01805**	**0.01742**	0.01627	**0.03650**	0.000770
TDDM	0.13729	0.15160	0.03881	0.15748	0.006640
TDDMv	0.03388	0.03440	0.03839	0.07866	0.005797
IDM	0.10742	0.14532	0.12836	0.13562	0.016445
PSD	0.02688	0.02694	**0.01451**	0.04262	**0.000699**

Method	Count W_1	MMD	$\mathrm{MMD}_{\text{train-scale}}$	JMMD^2 ($\times 10^3$)	ICMMD^2
EFDM	2.078	0.0542	0.0554	**0.167**	**0.0249**
FDDM	**1.222**	**0.0448**	**0.0426**	0.527	0.0292
TDDM	16.391	0.2574	0.3537	22.110	0.8821
TDDMv	11.677	0.0665	0.0844	5.188	0.4307
IDM	1.470	0.3305	0.3582	5.426	0.3750
PSD	2.052	0.0583	0.0640	0.379	0.0297

Table 5: Bandwidth sensitivity for bent two-Gaussian data. The columns vary the common training-derived bandwidth $\sigma = 0.133825$. JMMD^2 is displayed after multiplication by 10^3. All scores are lower-is-better. Bold marks the lowest value at the displayed precision, including ties.

Method	0.5σ	σ	2σ
EFDM	**0.133**	**0.167**	**0.163**
FDDM	0.385	0.527	0.568
TDDM	9.072	22.110	31.163
TDDMv	2.388	5.188	7.201
IDM	6.549	5.426	2.133
PSD	0.334	0.379	0.357

Count–separation visualization. For Figure 2, let $r_i = y_i - x_i^2$ be the vertical coordinate after the inverse transformation. We use the same uncentered moment estimator for real and generated sets:

$$\widehat{\Delta}(S) = 2\sqrt{\max\left\{\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\sum_{i=1}^{n} r_i^2,\, 0\right\}}. \tag{12}$$

The plotted quantity is an estimate from observed coordinates, not the latent component separation. Trend curves average $\widehat{\Delta}$ with Gaussian weights $w_i(n) = \exp[-(n_i - n)^2/(2h^2)]$, where n_i is the cardinality of set i and $h = 5$ count units for every method. Curves are shown only within each collection's observed count range and where $(\sum_i w_i)^2 / \sum_i w_i^2 \geq 20$. All scatter points are retained, including the eight PSD samples outside $[30, 110]$. The same real-test samples and trend are used in all panels.

Spatial summaries and bandwidth sensitivity. The upper panel of Table 4 reports the five spatial-summary errors. Table 5 varies the JMMD bandwidth to 0.5σ, σ, and 2σ, leaving the trained models and generated sets unchanged. EFDM has the lowest JMMD^2 in all three settings.

A.2 Implementation Details

For the trip, earthquake, and bent synthetic experiments, EFDM uses discrete-time DDPM diffusion with the existence-weighted objective in Eq. (4). QM9 uses continuous-time VP diffusion with predictor–corrector sampling. The configurations are detailed below.

Checkpoint selection. For the trip, earthquake, and bent synthetic experiments, we select each model's exponential moving average (EMA) checkpoint by the lowest validation loss.

Diffusion schedules. For binary EFDM on trip, earthquake, and bent synthetic data, we use a discrete DDPM with $T = 1{,}000$ and the linear variance schedule

$$\beta_t = 10^{-4} + \frac{t-1}{999}(0.02 - 10^{-4}), \qquad t = 1, \ldots, 1{,}000.$$

Generation uses 1,000 reverse predictor steps without a corrector. The binary EFDM sampler retains Gaussian variance β_1 in the final reverse step. The other non-PSD methods also use 1,000 discrete coordinate-diffusion steps; PSD uses its native 100-step thinning/superposition sampler. QM9 instead uses continuous-time VP diffusion with $\beta(t) = 0.1 + 19.9t$ for $t \in [0.001, 1]$ during training. Its predictor–corrector evaluation is described below.

TDDM deletion schedules. For the base TDDM on trip, earthquake, and bent synthetic data, we follow the sample-independent rate design of Campbell et al. (2023). Let $n_{\max}^{\text{train}} = \max_{S \in \mathcal{D}_{\text{train}}} |S|$. The shared forward deletion rate is calibrated to a cumulative intensity of $n_{\max}^{\text{train}} - 1$ over the forward process. Its rate is not adapted to each sample's initial count.

For TDDMv on these datasets, the forward deletion schedule has $T = 1{,}000$ steps. Given the initial cardinality n, its per-step deletion rate is

$$\lambda_t(n) = \begin{cases} 0, & 1 \le t \le 100 \text{ or } 901 \le t \le 1{,}000, \\ \dfrac{\max(n-1, 0)}{800}, & 101 \le t \le 900. \end{cases}$$

Thus deletion is disabled for the first and last 10% of steps. The initial cardinality is observed during training and replaced by the model's current prediction during generation.

For both variants, given an initial cardinality n, the forward count is $n_t = \max(n - D_t, 1)$, where $D_t \sim \text{Poisson}(\sum_{s=1}^{t} \lambda_s)$. This truncation ensures that at least one point remains.

Point-set representation. For binary EFDM, a training point set with cardinality $n \le N$ is embedded in N slots. We encode active and inactive slots with finite existence logits corresponding to probabilities $1 - \delta$ and δ, respectively, where $\delta \in (0, 1/2)$. We set $\delta = 0.003$ for the trip, earthquake, and bent synthetic datasets. This encoding preserves a strong separation between active and inactive slots while keeping the logits finite:

$$u_i^{(0)} = \begin{cases} \log \dfrac{1-\delta}{\delta}, & \text{active}, \\ -\log \dfrac{1-\delta}{\delta}, & \text{inactive}. \end{cases}$$

The inactive coordinates are sampled from the observed points in the same set. This yields a fixed number of slots while encoding the original cardinality through the existence variables.

Attribute encodings. For the shared-existence representation in Section 4.2.2, finite log-ratio coordinates for categorical labels can be obtained using a small amount of label smoothing. Active conditional labels can be encoded as

$$\rho_{ik}^{(m)} = (1-\delta)\mathbf{1}\{a_i^{(m)} = k\} + \frac{\delta}{K_m}.$$

Inactive slots use uniform conditional probabilities as auxiliary placeholders. The direct single-attribute parameterization used for QM9 and its relation to these coordinates are described in Appendix A.3.

Maximum cardinality. In our experiments, we choose EFDM's maximum slot number N based on the dataset cardinality range. For the trip dataset, we use $N = 400$, which is at least as large as the maximum observed cardinality. We use $N = 560$ for the earthquake dataset, $N = 29$ for QM9 molecules, and $N = 110$ for the bent synthetic benchmark. Subject to available computational and memory budgets, N can be set as large as the task requires; it need not be close to the maximum cardinality in the training data.

Model configurations. For the trip and earthquake datasets, EFDM, FDDM, TDDM, TDDMv, and PSD use Transformer-based architectures (Vaswani et al., 2017) with the dimensions listed in Table 6. IDM uses a point-wise feedforward network, since it models points independently and does not use interactions among points. TDDM and TDDMv use fused width 128 on both datasets.

Table 6: Model configurations for the trip and earthquake experiments. EFDM, FDDM, TDDM, TDDMv, and PSD use Transformer-based architectures. IDM uses a point-wise MLP.

Dataset	Method	Backbone	Emb. Dim	FF/Hidden Dim	Heads	Layers
Trip	IDM	MLP	128	256	–	8
Trip	EFDM/FDDM/TDDM/TDDMv/PSD	Transformer	128	256	4	8
Earthquake	IDM	MLP	128	256	–	8
Earthquake	EFDM/FDDM/TDDM/TDDMv/PSD	Transformer	128	256	4	8

For QM9, EFDM uses an EGNN backbone (Satorras et al., 2021) and a Transformer component (Vaswani et al., 2017), configured as shown in Table 7. Existence probabilities are used

in the fusion branch but are not supplied as explicit inputs to the EGNN. The model is trained with batch size 64 and learning rate 3×10^{-5}. All QM9 baseline results in the main paper are taken from the literature; these methods are not retrained for the comparison.

Table 7: Components of the EFDM model used for the QM9 molecule experiment.

Component	Hidden/Embedding Dim	Heads	Layers
EGNN	256	–	9
Transformer	256	8	4

QM9 training and label encoding. The QM9 EFDM training run spans 8,000 epochs, corresponding to 800 million processed molecule instances. EMA used a half-life of 500,000 processed molecules with a ramp-up ratio of 0.05; at batch size 64, its asymptotic decay is $2^{-64/500000} \approx 0.9999113$ per update. In the direct categorical representation, the inactive reference logit is fixed at $U_{i0} = 0$. For an active slot, the observed atom-type logit is $+1$ and every other type logit is -1; every type logit is -1 for an inactive slot. The binary-existence smoothing parameter $\delta = 0.003$ above does not define QM9 type labels.

QM9 checkpoint and evaluation. For the reported QM9 result, we use the EMA weights from the last training epoch (epoch 8,000). We evaluate this checkpoint once by generating 10,000 molecules in batches of 64, rather than averaging over repeated runs. Atom stability is 99.04%, molecule stability is 91.18%, and validity is 96.38%. Across six late EMA checkpoints (epochs 6,000–8,000), atom stability ranges from 98.99% to 99.10%, molecule stability from 90.23% to 91.38%, and validity from 95.91% to 96.38%; every checkpoint exceeds the FDDM and TDDM scores on all three metrics. We use predictor–corrector sampling. With diffusion time normalized to $T = 1$, the predictor step size is 0.05 for $t > 0.5T$ and 0.001 thereafter. Corrector updates are enabled only for $0.003 \le t < 0.1T$, with five Langevin updates per predictor step and a signal-to-noise ratio (SNR) of 0.3. The sampler uses 510 predictor updates and 485 corrector updates, for a total of 995 network function evaluations (NFE). No noise is added in the final predictor update.

A.3 Single-Attribute Parameterization and Readout

For a single attribute with K meaningful categories, the shared-existence representation in Section 4.2.2 uses $(x_i, u_i, v_{i2}, \ldots, v_{iK})$, with $v_{i1} = 0$. An alternative parameterization fixes the logit of the inactive category to zero and represents the entire categorical field directly:

$$\begin{aligned} U_{i0} = 0, \qquad U_{ik} = \log \frac{\gamma_{ik}}{\gamma_{i0}}, \quad k = 1, \ldots, K, \\ \boldsymbol{\gamma}_i = \mathrm{softmax}(0, U_{i1}, \ldots, U_{iK}). \end{aligned} \tag{13}$$

Both parameterizations have K free coordinates and represent every categorical distribution in the simplex interior. The QM9 experiments use the direct representation, jointly diffusing $(x_i, U_{i1}, \ldots, U_{iK})$.

Equivalence of the categorical fields. Suppress the slot index and write $Z_v = \sum_{k=1}^{K} \exp(v_k)$, where $v_1 = 0$. The map from shared-existence coordinates to direct logits is

$$U_k = u + v_k - \log Z_v, \qquad k = 1, \ldots, K. \tag{14}$$

Indeed, $\sum_k \exp(U_k) = \exp(u)$, so

$$\gamma_0 = \frac{1}{1 + \exp(u)} = 1 - \sigma(u), \qquad \gamma_k = \frac{\exp(u)}{1 + \exp(u)} \frac{\exp(v_k)}{Z_v} = \sigma(u)\rho_k. \tag{15}$$

Conversely, any $\boldsymbol{U} \in \mathbb{R}^K$ determines

$$u = \log \sum_{k=1}^{K} \exp(U_k), \qquad v_k = U_k - U_1, \qquad \rho_k = \exp(U_k - u). \tag{16}$$

These maps are inverses: substituting the latter expression gives $\log Z_v = u - U_1$ and recovers every U_k in Eq. (14). Thus a categorical draw from $\boldsymbol{\gamma}$ has exactly the same distribution as drawing presence from $\mathrm{Bernoulli}(\sigma(u))$ and, when present, drawing the attribute from $\mathrm{Categorical}(\boldsymbol{\rho})$.

The two parameterizations represent the same categorical distributions and probabilistic readout. For $K > 1$, independent Gaussian diffusion in these nonlinearly related coordinates generally yields different dynamics and training objectives.

Deterministic readout in QM9. QM9 trains these direct logits with the fixed ± 1 encoding in Appendix A.2 and decodes the recovered full field by argmax:

$$\hat{c}_i = \underset{k \in \{0,\dots,K\}}{\arg\max}\, \hat{U}_{ik}, \qquad \hat{U}_{i0} = 0. \tag{17}$$

Category 0 discards the slot; otherwise, the slot is retained with attribute $\hat{c}_i$. With a consistent tie-breaking convention, the corresponding MAP decision is identical in either parameterization because the full categorical probabilities agree. Away from ties, retention is equivalent to

$$\max_k \hat{U}_{ik} > 0 \quad \Longleftrightarrow \quad \hat{e}_i \max_k \hat{\rho}_{ik} > 1 - \hat{e}_i. \tag{18}$$

This deterministic rule is generally different from either probabilistic readout or thresholding the total existence probability at $1/2$.

A.4 Evaluation Metric Details

Let P and Q denote the real and generated distributions over point sets $S \subset \mathbb{R}^d$. Write $n = |S|$ for the cardinality of a point set and $P_{\mathrm{card}}, Q_{\mathrm{card}}$ for the corresponding marginal distributions. Let $\mathcal{R} = (R_i)_{i=1}^r$ and $\mathcal{G} = (G_j)_{j=1}^g$ be the real test and generated collections, and let $\mathcal{T} = (T_a)_{a=1}^t$ be the training collection. Throughout, a collection contains point sets, rather than individual points. The kernel estimators below require $r, g \geq 2$ and positive bandwidths.

Cardinality and spatial summaries. Cardinality error is $W_1(\widehat{P}_{\mathrm{card}}, \widehat{Q}_{\mathrm{card}})$, the one-dimensional Wasserstein distance between the empirical count distributions. For a nonempty set $S = \{x_i\}_{i=1}^n$, coordinate a, and moment order h, define

$$\bar{x}_a = \frac{1}{n}\sum_{i=1}^n x_{ia}, \qquad m_{h,a} = \frac{1}{n}\sum_{i=1}^n (x_{ia} - \bar{x}_a)^h. \tag{19}$$

The coordinate-wise summaries are

$$\mathrm{Mean}_a = \bar{x}_a, \quad \mathrm{Var}_a = m_{2,a}, \quad \mathrm{Skew}_a = \frac{m_{3,a}}{m_{2,a}^{3/2}}, \quad \mathrm{Kurt}_a = \frac{m_{4,a}}{m_{2,a}^2} - 3. \tag{20}$$

Variance divides by n, not $n-1$. Skew and Kurt are the uncorrected standardized third central moment and Fisher excess kurtosis, respectively; the formulas above define them when coordinate variance is positive. For numerical stability, the reported implementation instead uses standardized coordinates $z_i = (x_{ia} - \bar{x}_a)/(s_a + 10^{-8})$ when the coordinate standard deviation $s_a = \sqrt{m_{2,a}}$ is at least 10^{-8}, and sets all $z_i = 0$ when $s_a < 10^{-8}$. It then computes Skew as $n^{-1}\sum_i z_i^3$ and Kurt as $n^{-1}\sum_i z_i^4 - 3$. Thus coordinates with $s_a < 10^{-8}$ contribute skewness 0 and Fisher excess kurtosis -3. Singleton sets remain in the spatial-summary distributions. The mean nearest-neighbor distance, for $n \geq 2$, is

$$\mathrm{NN1}(S) = \frac{1}{n}\sum_{i=1}^n \min_{j \neq i} \|x_i - x_j\|_2. \tag{21}$$

Only observed points enter these summaries; inactive or padding slots are excluded. For a coordinate-wise summary f, the reported discrepancy is

$$\mathcal{W}_f(\mathcal{R}, \mathcal{G}) = \frac{1}{d}\sum_{a=1}^d W_1\left(\frac{1}{r}\sum_{i=1}^r \delta_{f_a(R_i)}, \frac{1}{g}\sum_{j=1}^g \delta_{f_a(G_j)}\right). \tag{22}$$

This compares per-set feature distributions along each coordinate before averaging the distances; it does not pool the individual points. For NN1, the analogous formula has one scalar feature and no coordinate average.

For the Earthquake and Trip experiments, empty generated proposals are rejected and resampled until 4,096 nonempty sets are obtained per method and split. All ten reported metrics, including Count W_1 and the spatial-summary discrepancies in Eq. (22), use this same accepted generated collection. NN1 is set to zero for singleton sets. All 1,024 generated sets per method in the bent synthetic experiment are also nonempty.

Sliced distance and normalization. For nonempty sets, spatial MMD uses the 50-projection sliced-W_2 distance (Lüdke et al., 2025):

$$D_{50}(S,S') = \left[\frac{1}{50}\sum_{\ell=1}^{50} W_2^2\big((\theta_\ell^\top)_\#\mu_S, (\theta_\ell^\top)_\#\mu_{S'}\big)\right]^{1/2}, \quad \mu_S = \frac{1}{|S|}\sum_{x\in S}\delta_x. \tag{23}$$

Here θ_ℓ are random unit projection directions and μ_S is the uniformly weighted empirical point measure.

Spatial-summary W_1 uses coordinates standardized with training-set statistics for Earthquake and Trip. For the bent synthetic benchmark, it uses the observed bent coordinates without standardization or inverse transformation. The MMD, calibrated MMD, and train-scale MMD evaluations reverse this standardization, map the supplied spatial bounds to $[-1,1]^d$, and clip coordinates to this domain. JMMD and ICMMD use the bounded coordinates for Earthquake, but use unclipped standardized coordinates for Trip. The bent synthetic benchmark uses training-derived bounds and the same bounded mapping and clipping as Earthquake for all MMD-family metrics. Its structural diagnostics instead apply the inverse bending transform to both real and generated points. Their implementations do not normalize inputs internally. Within each dataset and split, the same coordinate transformation is applied to training, test, and generated sets for a given metric. A three-seed aggregate is reported only when all three scores are finite. The official PSD Wasserstein path draws projections without passing a fixed seed to each distance call. JMMD and ICMMD instead reuse the same integer projection seed for every pair, ensuring the same 50 directions in a common ambient dimension. For equal-cardinality nonempty sets, their distance has the equivalent form

$$D_{50}(S,S')^2 = \frac{1}{50n}\sum_{\ell=1}^{50}\sum_{i=1}^{n}(s_{\ell,(i)} - s'_{\ell,(i)})^2, \tag{24}$$

where $s_{\ell,(i)}$ is the ith sorted projection of S along θ_ℓ. Thus κ_σ is a Gaussian kernel on the sorted projected-coordinate representation within each cardinality. The implementation sets $D_{50}(\emptyset,\emptyset) = 0$ for these two metrics; unequal-count pairs have zero joint kernel and do not require a distance calculation. Finitely many projections do not in general identify every distribution of the original point sets.

MMD estimators. We use the MMD framework of Gretton et al. (2012), with a Wasserstein-distance kernel for spatial point sets as in PSD (Lüdke et al., 2025). The metric labeled MMD uses the unsquared sliced distance from Equation (23):

$$k_\sigma(S,S') = \exp\left[-\frac{D_{50}(S,S')}{2\sigma^2}\right].$$

For a kernel k, define the biased estimator, including self-pairs, by

$$\widehat{D}_{v,k}(\mathcal{R},\mathcal{G}) = \frac{1}{r^2}\sum_{i,i'=1}^{r} k(R_i,R_{i'}) + \frac{1}{g^2}\sum_{j,j'=1}^{g} k(G_j,G_{j'}) - \frac{2}{rg}\sum_{i=1}^{r}\sum_{j=1}^{g} k(R_i,G_j). \tag{25}$$

MMD reports $\sqrt{\widehat{D}_{v,k_\sigma}}$. For its default bandwidth, form the concatenation (with multiplicities)

$$\begin{aligned}\mathcal{D}_{\mathrm{eval}} &= \big(D_{50}(R_i,R_{i'})\big)_{i,i'} \,\|\, \big(D_{50}(G_j,G_{j'})\big)_{j,j'} \,\|\, \big(D_{50}(R_i,G_j)\big)_{i,j},\\ \sigma_{\mathrm{eval}} &= \operatorname{median}\mathcal{D}_{\mathrm{eval}}.\end{aligned} \tag{26}$$

Each within-collection block contains all ordered pairs, including its diagonal, and the cross block is included once. This is the bandwidth rule of the official PSD implementation (Lüdke et al., 2025), which may yield a different bandwidth for each generated collection. Calibrated MMD retains the distance routine, kernel k_σ, and square-root biased estimator, but fixes its bandwidth from the training collection:

$$\sigma_{\text{train}} = \text{median}\{D_{50}(T_i, T_j) : 1 \le i, j \le t\}. \tag{27}$$

The median includes all t^2 ordered pairs, including self-pairs, and is fixed across methods for a given training split without using test or generated samples to select it.

For train-scale MMD, let $m = \sigma_{\text{train}}$ denote the same median training distance and instead use

$$k_{\text{train-scale}}(S, S') = \exp\left[-\frac{D_{50}(S, S')}{2m}\right]. \tag{28}$$

The metric still reports the square root of the biased MMD^2 estimator, but its kernel denominator is $2m$, rather than $2m^2$ as in calibrated MMD. Equivalently, the squared kernel bandwidth is m. These are distinct metrics and are labeled separately. The implementation uses 50 fresh random projections per ordered pair and full real–real, generated–generated, and cross-collection grids. Both MMD variants use all 4,096 generated sets against 300 test sets for Earthquake and 91 test sets for Trip. The bent synthetic evaluation uses all 1,024 generated and 2,000 test sets.

The unbiased squared-MMD estimator removes only within-collection self-pairs (Gretton et al., 2012):

$$\widehat{D}_{u,k}(\mathcal{R}, \mathcal{G}) = \frac{\sum_{i \ne i'} k(R_i, R_{i'})}{r(r-1)} + \frac{\sum_{j \ne j'} k(G_j, G_{j'})}{g(g-1)} - \frac{2}{rg}\sum_{i,j} k(R_i, G_j). \tag{29}$$

For joint cardinality and spatial comparisons, define the product kernel (Long et al., 2017)

$$K_\sigma((n, S), (n', S')) = \mathbf{1}_{\{n=n'\}}\kappa_\sigma(S, S'), \qquad \kappa_\sigma(S, S') = \exp\left[-\frac{D_{50}(S, S')^2}{2\sigma^2}\right]. \tag{30}$$

JMMD reports $\widehat{D}_{u,K_\sigma}$, a signed estimate of $\text{MMD}^2_{K_\sigma}(P_{\text{card},S}, Q_{\text{card},S})$. The denominators are the total pair counts, not the numbers of pairs with equal cardinality. For a fixed kernel and independent samples, this is an unbiased estimate of the population squared discrepancy. It is reported with its sign, without a square root or clipping. Negative estimates can occur through finite-sample fluctuations; their sign alone does not establish statistical indistinguishability.

Training bandwidth for JMMD and ICMMD. Let

$$\begin{aligned} \mathcal{P}_{\text{eq}} &\subseteq \{(a, b) : 1 \le a < b \le t,\ |T_a| = |T_b|\}, \\ \sigma_{\text{eq}} &= \text{median}\{D_{50}(T_a, T_b) : (a, b) \in \mathcal{P}_{\text{eq}}\}. \end{aligned} \tag{31}$$

All distinct unordered equal-count training pairs are used if there are at most 20,000; otherwise 20,000 are sampled uniformly without replacement. The pairs are pooled across counts, so count groups are weighted by their numbers of pairs in this bandwidth calculation. Self-pairs are excluded. The resulting bandwidth is shared by JMMD and ICMMD and fixed across methods for the same split and projection seed. Sensitivity evaluations at $0.5\sigma_{\text{eq}}$ and $2\sigma_{\text{eq}}$ can reuse the same distances. This convention differs from the all-ordered-pair training median used for calibrated MMD. The reported Earthquake and Trip evaluations use the training equal-count bandwidth with a fixed projection seed, shared across methods within a split. For Earthquake, the bandwidths are approximately 0.1730, 0.1750, and 0.1692 across the three split/training seeds.

Joint versus conditional discrepancies. The product-kernel construction follows joint mean embeddings (Long et al., 2017). Write $P_n = P(S \mid |S| = n)$ and $Q_n = Q(S \mid |S| = n)$ wherever the corresponding count probabilities are positive. If $p_n = P_{\text{card}}(n)$, $q_n = Q_{\text{card}}(n)$, and μ_{P_n}, μ_{Q_n} are the spatial kernel mean embeddings within cardinality n, then

$$\text{JMMD}^2(P, Q) = \sum_n \|p_n\mu_{P_n} - q_n\mu_{Q_n}\|^2_{\mathcal{H}_n}. \tag{32}$$

A weighted embedding is defined to be zero when its count probability is zero, so this expression does not require a conditional law at an absent count. It therefore compares both the count probabilities

and the associated spatial distributions. ICMMD instead integrates the conditional discrepancies against the real cardinality marginal:

$$\mathrm{ICMMD}^2(P, Q; P_{\mathrm{card}}) = \sum_{n: P_{\mathrm{card}}(n) > 0} P_{\mathrm{card}}(n)\, \mathrm{MMD}^2_{\kappa_\sigma}(P_n, Q_n), \tag{33}$$

This population target requires Q_n to be defined on the support of P_{card}. This is a discrete-cardinality adaptation of integrated conditional MMD (Yan et al., 2024); that reference studies continuous conditioning variables and kernel-smoothed estimators. Our exact-count eligibility, weighting, and penalty rules are specified below.

Reported ICMMD score and coverage. Write $\mathcal{R}_n = \{R_i : |R_i| = n\}$, $\mathcal{G}_n = \{G_j : |G_j| = n\}$, $r_n = |\mathcal{R}_n|$, and $g_n = |\mathcal{G}_n|$. Using the implementation's default minimum real group size of two, define

$$\mathcal{E} = \{n : r_n \geq 2\}, \qquad \mathcal{C} = \{n \in \mathcal{E} : g_n \geq 2\}, \qquad w_n = \frac{r_n}{\sum_{a \in \mathcal{E}} r_a}. \tag{34}$$

Eligibility and weights depend only on the real test counts. For a collection $A = (A_i)_{i=1}^s$ with $s \geq 2$, let

$$U_\sigma(A) = \frac{2}{s(s-1)} \sum_{i < i'} \kappa_\sigma(A_i, A_{i'}). \tag{35}$$

The reported score is

$$\widehat{\mathrm{ICMMD}}^2 = \sum_{n \in \mathcal{E}} w_n \widehat{\Delta}_n, \qquad \widehat{\Delta}_n = \begin{cases} \widehat{D}_{u,\kappa_\sigma}(\mathcal{R}_n, \mathcal{G}_n), & n \in \mathcal{C}, \\ 1 + U_\sigma(\mathcal{R}_n), & n \in \mathcal{E} \setminus \mathcal{C}. \end{cases} \tag{36}$$

The second case applies to both absent generated counts and generated groups containing only one set. Since $0 \leq \kappa_\sigma \leq 1$, $1 + U_\sigma(\mathcal{R}_n)$ is a conservative upper bound for a covered group's squared-MMD estimate. Using it as a penalty prevents insufficient generated support from being silently omitted. The complete restricted and penalized score is not an unbiased estimator of the unrestricted population target. Covered-group estimates retain their signs without clipping.

To make the penalty's contribution visible, the implementation also returns

$$\mathrm{EligibleMass} = \frac{\sum_{n \in \mathcal{E}} r_n}{r}, \qquad \mathrm{Coverage} = \sum_{n \in \mathcal{C}} w_n, \tag{37}$$

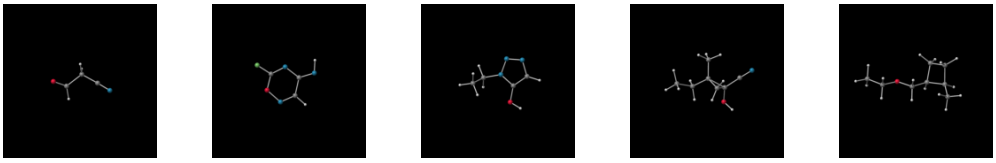
$$\widehat{\mathrm{ICMMD}}^2_{\mathrm{covered}} = \frac{\sum_{n \in \mathcal{C}} w_n \widehat{\Delta}_n}{\sum_{n \in \mathcal{C}} w_n}. \tag{38}$$

The main score and coverage are normalized by eligible real mass; the covered score is normalized by covered real mass. The covered score is undefined when coverage is zero, and the evaluation is undefined if no real count is eligible.

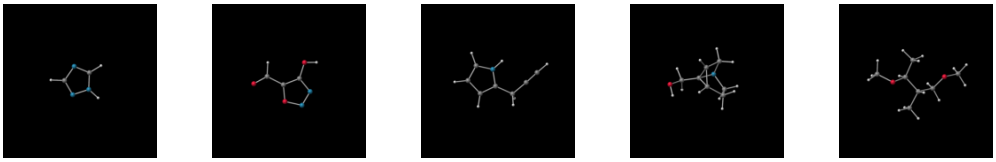
A.5 Spatial evaluation protocols and supplementary results

Runs and checkpoint selection. The trip and earthquake comparisons use three split/training seeds. Within a dataset and seed, all methods share the same data split; changing the seed changes both the split and training randomness. Runs are trained for 60,000 epochs, and each reported checkpoint is selected by the lowest validation loss evaluated with EMA weights. Test metrics are not used to select checkpoints. Earthquake models use Adam (Kingma and Ba, 2015). EFDM and FDDM use batches of 64 point sets, TDDM and TDDMv use 32 sets, PSD uses 128 sets, and IDM uses 2,048 individual points. The learning rate is 10^{-4} with zero weight decay except for PSD, which uses 10^{-3} and weight decay 10^{-4}, with gradient clipping at 2. All Earthquake methods use EMA decay 0.9999 per update; the lowest-loss EMA checkpoint is selected from validation evaluations every 300 epochs (200 candidates).

For trip, each split contains 268 training, 89 validation, and 91 test point sets. Each day's point set is subsampled to approximately 10% of its points. The density filter and coordinate normalization are fitted on the training split. Trip uses an EMA half-life of 100 epochs for EFDM, IDM, and PSD, and 500 epochs for FDDM, TDDM, and TDDMv, with ramp-up parameter 0.2. Trip uses Adam (Kingma and Ba, 2015) with learning rate 10^{-4} and zero weight decay for non-PSD methods; PSD uses learning rate 10^{-3} and weight decay 10^{-4}. Training batches are 64 sets for EFDM and FDDM, 32 sets for TDDM and TDDMv, 128 sets for PSD, and 2,048 points for IDM.



(a) Real data (QM9 molecules)



(b) EFDM samples

Figure 3: Examples of molecules from the QM9 dataset (top) and samples generated by EFDM (bottom). From left to right, each column shows molecules with 8, 10, 15, 20, and 25 atoms, respectively. Bonds are inferred from interatomic distances and are shown only for visualization.

Sampling and aggregation. Trip uses 4,096 generated point sets per method and seed against all 91 test sets, sampled in batches of 32 from the same selected EMA checkpoints. Its samplers use 1,000 predictor steps for non-PSD methods and PSD's native 100 steps, without a corrector. Earthquake uses 4,096 generated sets against all 300 test sets for all ten main-table metrics, including both MMD variants. Both datasets report mean $\pm$ sample standard deviation across three split/training seeds (divisor $3 - 1$), computed directly from unrounded per-seed metric values. These standard deviations reflect split and training variability; they are not standard errors or confidence intervals. Training budgets are matched by epoch count, not by computational cost.

The implementation names `mask`, `nogroup`, `jump`, and `jump_adaptive` correspond to FDDM, IDM, TDDM, and TDDMv, respectively. FDDM and IDM sample cardinalities from a distribution fitted to training counts; the other methods generate their own cardinalities.

Trip MMD saturation. All ten Trip metrics use the same 4,096 generated sets against all 91 test sets. MMD scores lie close to the diagonal-kernel value $\sqrt{1/4096 + 1/91} \approx 0.105987$, consistent with saturation. The small differences in this column are not interpreted as evidence of better generation quality. Both MMD and train-scale MMD remain in the main table; they use the distinct bandwidth conventions in Appendix A.4.

A.6 Additional Visualizations

This section provides additional visualizations that complement the quantitative results in the main paper. The figures show molecule examples for QM9, together with cardinality summaries and point-set examples for the Earthquake and Trip datasets.

QM9 molecules. Figure 3 compares molecules from QM9 with EFDM-generated samples with matching atom counts. The examples include both chain-like and cyclic configurations.

Spatial datasets. Figure 4 summarizes nonempty generated collections for Earthquake and Trip, generated without a corrector and summarized over three split/training seeds. TDDM's count distribution is shifted toward larger sets on both datasets, while TDDMv concentrates on a narrower range than the real data. The coverage panels quantify whether individual eligible counts are represented, complementing the overall count-distribution comparison.

Figures 5 and 6 show point-set examples in training-defined count ranges. Annotated cardinalities show both matched counts and remaining differences within a range; the high-count examples do not

all have equal cardinality. Blank panels indicate that the displayed collection contains no sample in the corresponding range.

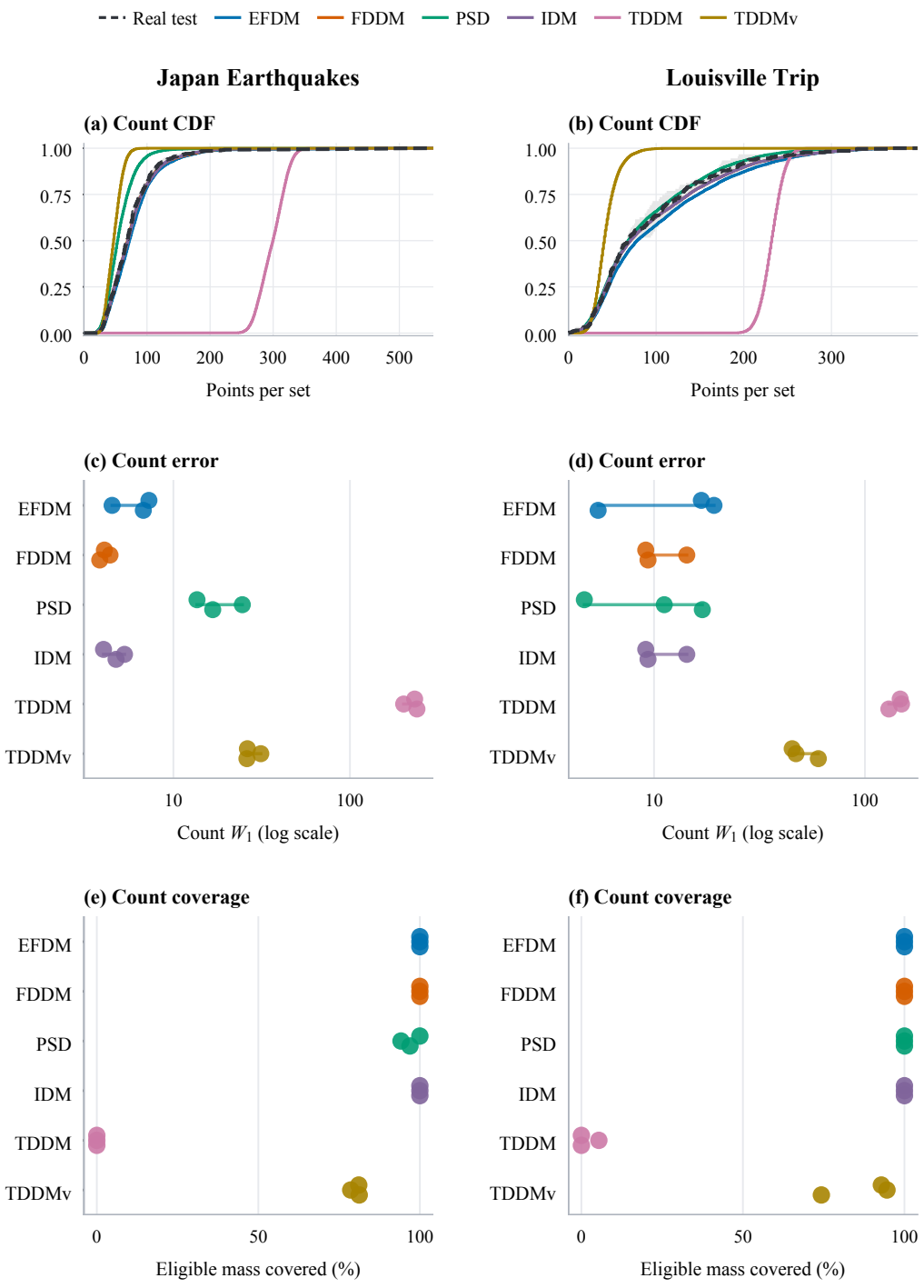


Figure 4: Cardinality summaries for Earthquake (left) and Trip (right), using 4,096 nonempty generated sets per method and split. Rows show count CDFs, count W_1, and coverage. CDFs are averaged over three splits; gray bands show the real split range. Dots denote individual splits. Coverage is the fraction of eligible test mass at counts with at least two generated sets; eligibility requires at least two real test sets at that count. Eligible mass is 86–87% for Earthquake and 31–41% for Trip.

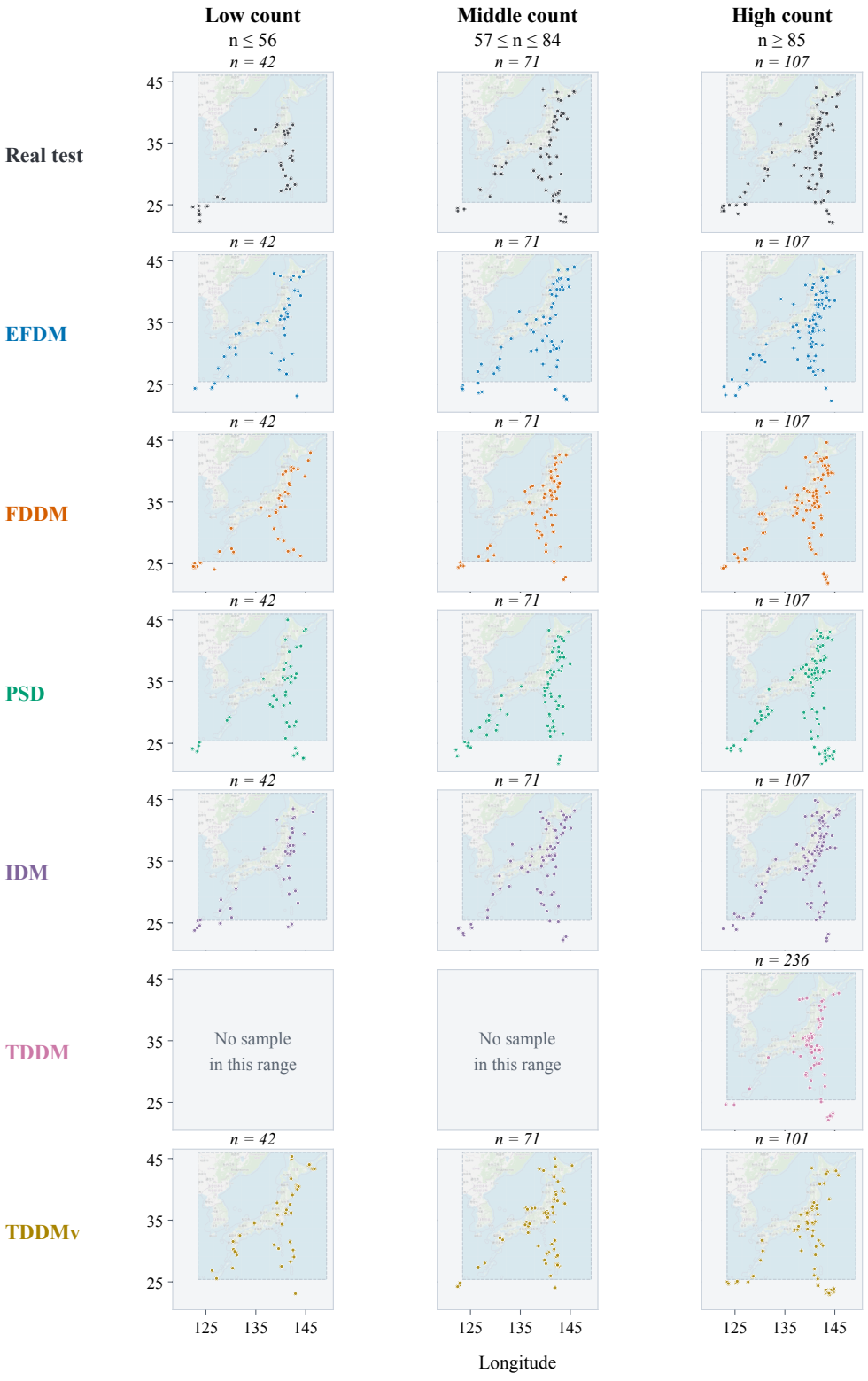


Figure 5: Japan Earthquakes: point-set examples with methods in rows and three training-defined count ranges in columns. Annotated counts show differences within each range. All panels share longitude–latitude axes; dashed boxes mark the map-background extent. Blank panels denote no sample in that range within the displayed collection.



Figure 6: Louisville Trip: pickup-location sets with methods in rows and three training-defined count ranges in columns. Annotated counts show differences within each range. Coordinates use training standardization, shared axes, and no per-set alignment. Blank panels denote no sample in that range within the displayed collection.