Flow Matching for Count Data

Ganchao Wei*
Department of Neurobiology
Department of Statistical Science
Duke University
Durham, NC, USA
weiganchao@gmail.com

John Pearson
Department of Neurobiology
Department of Electrical and Computer Engineering
Duke University
Durham, NC, USA
john.pearson@nyu.edu

Abstract

High-dimensional count data arise in applications such as single-cell RNA sequencing and neural spike trains, where mappings between distributions across successive batches or time points form critical components of data analysis. The recent success of diffusion- and flow-based deep generative models for images, video, and text motivates extending these ideas to count-valued settings, but many existing methods either treat each count as a categorical state or transform counts into a continuous space, neither of which is natural or efficient when the count range is large. We propose count-FM, a flow-matching framework for count data based on a continuous-time birth-death process with local unit jumps. Count-FM learns marginal transitions efficiently in count space through simulation-free training of conditional transition rates, allowing transport between arbitrary count-distributed source and target populations. In simulation, count-FM variants achieve strong sample quality while using substantially fewer parameters. We further apply count-FM to scRNA-seq and neural spike-train data for unconditional generation, transport, and conditional generation. Across these tasks, count-FM yields improved sample quality, greater modeling efficiency, and interpretable transport paths.

1 Introduction

High-dimensional count data arise in many scientific applications, including key biological data types such as single-cell RNA sequencing and neural spike trains. However, their discreteness, sparsity, and complex correlation structure make flexible joint modeling difficult, especially in high dimensions. Deep generative models provide a powerful framework for addressing this problem, with major developments including variational autoencoders [Kingma and Welling, 2014], generative adversarial networks [Goodfellow et al., 2014], normalizing flows [Kingma and Dhariwal, 2018], diffusion models [Ho et al., 2020, Song et al., 2021], and flow matching [Lipman et al., 2023]. In particular, diffusion- and flow-based methods have recently attracted substantial attention because of their strong empirical performance, flexible path-based formulations, and efficient training procedures [Ho et al., 2020, Song et al., 2021, Lipman et al., 2023]. This progress has motivated growing interest in extending these generative frameworks from continuous domains to discrete data.

For modeling discrete data, recent approaches include categorical diffusion models such as D3PM [Austin et al., 2021], tauLDR [Campbell et al., 2022], SEDD [Lou et al., 2024], and masked diffusion models [Sahoo et al., 2024, Shi et al., 2024, Chao et al., 2025], as well as discrete flow methods such as Dirichlet-FM [Stark et al., 2024], discrete-FM [Gat et al., 2024], and CTMC-based discrete flows [Campbell et al., 2024]. When adapted to count data, categorical-state methods typically require a predefined count range, and their categorical representation does not itself encode the ordering of counts. Another approach maps counts into a continuous space through dequantization or continuous

*Corresponding author. Alternative email: gwei@flatironinstitute.org

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

representations [Ho et al., 2019, Hoogeboom et al., 2021], but this can blur the underlying discrete structure [Hoogeboom et al., 2019, Luo et al., 2021]. Count-specific alternatives include Poisson-JUMP [Chen and Zhou, 2023], Count Bridges [Fishman et al., 2026], and ItDPDM [Bhattacharya et al., 2025]. Poisson-JUMP and ItDPDM are not designed for transport between arbitrary count distributions, while Count Bridges learns an endpoint posterior and exact finite-time bridges.

To address these challenges while retaining the efficient training and generation of flow matching, we propose count-FM, a flow-matching framework for count data based on a continuous-time birth-death process with local unit jumps. We model transport directly in count space through coordinate-wise local births and deaths, so that transitions respect the geometry of counts and remain count-valued along the path. The conditional binomial bridge yields an efficient training objective and enables tractable training of the marginal transition rates. The model is also parameter-efficient, since it predicts only local birth and death rates rather than a full categorical distribution over all count levels, without requiring a predefined maximum count. This makes count-FM well suited to high-dimensional count data with large count ranges.

We first validate count-FM on a two-dimensional simulation before applying it to single-cell RNA-seq for unconditional generation and transport, followed by conditional generation of brain data, multiregion hippocampal and entorhinal spike trains. Our main contributions are as follows:

- We propose count-FM, a flow-matching framework for count data based on a continuous-time birth-death process with local unit jumps, so that transport is modeled directly in count space and intermediate states remain count-valued along the path.
- We develop a tractable training scheme for marginal transition rates through a conditional binomial bridge while retaining the efficient training and generation of flow matching.
- We obtain a parameter-efficient formulation that predicts only local birth and death rates rather than a full categorical distribution over all count levels, making it well suited to high-dimensional count data with large count ranges.

The Python implementation of count-FM is available at `https://github.com/weigcdsb/count-flow-matching`.

2 Method

In this section, we first formulate count-FM and derive its training objective, then describe sampling, endpoint coupling, and conditional generation. Appendix A.4 analyzes error from endpoint truncation, rate estimation, and sampling discretization.

Here, we consider a pair of counting distributions on d variables, $\mathbb{N}_0^d = \{0, 1, 2, \dots\}^d$, where $x_0 \sim p_0$ is a source sample and $x_1 \sim p_1$ is a target sample. We model the transition between these two distributions as a continuous-time Markov jump process (CTMC) [Campbell et al., 2022, 2024] with local unit births and deaths. Let $X_t \in \mathbb{N}_0^d$ denote the count vector at time t, and let $e_i \in \mathbb{R}^d$ denote the ith standard basis vector, that is, the vector whose ith entry is 1 and all other entries are 0. Over a small time interval $h > 0$, the process evolves according to

$$\mathbb{P}(X_{t+h} = x + e_i \mid X_t = x) = h\,\lambda_{t,i}(x) + o(h),$$
$$\mathbb{P}(X_{t+h} = x - e_i \mid X_t = x) = h\,\mu_{t,i}(x) + o(h),$$

and the probability of staying at x is

$$\mathbb{P}(X_{t+h} = x \mid X_t = x) = 1 - h\sum_{i=1}^{d}\big(\lambda_{t,i}(x) + \mu_{t,i}(x)\big) + o(h).$$

Death is disallowed when $x^{(i)} = 0$. The goal is to learn the time-dependent birth rates $\lambda_t(x)$ and death rates $\mu_t(x)$.

One benefit of this local birth-death parameterization is parameter efficiency for count-valued data. Many categorical-state discrete diffusion and flow models for count data represent each coordinate over all count levels up to a maximum count, so the total output dimension is $\sum_{i=1}^{d}(C_i+1)$, where C_i denotes the maximum count for coordinate i. By contrast, count-FM parameterizes each coordinate

using only birth and death rates for local ± 1 jumps, so its rate output dimension is always $2d$, regardless of maximum counts. This is more important in high-count and high-dimensional settings, where categorical-state parameterizations grow increasingly expensive. Empirically, in both the simulation and scRNA experiments, the count-FM variants use fewer trainable parameters than all baseline models with available parameter counts (Tables 1 and 2).

2.1 Conditional count bridge and training objective

Let π be a coupling of p_0 and p_1. Given an endpoint pair $(x_0, x_1) \sim \pi$, we define the conditional bridge coordinatewise. For each coordinate i,

$$X_t^{(i)} = x_0^{(i)} + \operatorname{sgn}\big(x_1^{(i)} - x_0^{(i)}\big)\, B_t^{(i)}, \qquad B_t^{(i)} \sim \operatorname{Binomial}\Big(\Big|x_1^{(i)} - x_0^{(i)}\Big|, t\Big), \tag{1}$$

independently across coordinates given (x_0, x_1), where sgn is the sign function. This bridge defines a conditional probability path $p_t(x \mid x_0, x_1)$ whose coordinate-wise mean moves linearly from x_0 to x_1, with local count changes. Conditional on (x_0, x_1), each coordinate remains between its endpoints and evolves monotonically in expectation. After marginalization, however, the learned rates depend on the full state x, so generated trajectories need not be monotone and cross-coordinate dependence can be captured.

For a fixed coordinate i, let $x = X_t^{(i)}$ and write $p_t^{(i)}(x \mid x_0, x_1)$ for the corresponding one-dimensional bridge marginal. Then mass preservation (equivalently, the one-dimensional Kolmogorov forward equation for a local CTMC [Holderrieth et al., 2025]) gives

$$\begin{aligned}
\partial_t p_t^{(i)}(x \mid x_0, x_1) = &\underbrace{\lambda_{t,i}(x-1 \mid x_0, x_1)\, p_t^{(i)}(x-1 \mid x_0, x_1)}_{\text{birth inflow}} \\
&+ \underbrace{\mu_{t,i}(x+1 \mid x_0, x_1)\, p_t^{(i)}(x+1 \mid x_0, x_1)}_{\text{death inflow}} \\
&- \underbrace{\big(\lambda_{t,i}(x \mid x_0, x_1) + \mu_{t,i}(x \mid x_0, x_1)\big) p_t^{(i)}(x \mid x_0, x_1)}_{\text{outflow from } x}.
\end{aligned} \tag{2}$$

This equation expresses local mass balance, where the change in mass equals influx minus outflux. Requiring the conditional rates to satisfy it ensures that the CTMC generates the conditional path $p_t(\cdot \mid x_0, x_1)$. Substituting the conditional binomial bridge (1) into (2) yields

$$\lambda_{t,i}(x \mid x_0, x_1) = \frac{\big(x_1^{(i)} - x\big)_+}{1-t}, \qquad \mu_{t,i}(x \mid x_0, x_1) = \frac{\big(x - x_1^{(i)}\big)_+}{1-t},$$

where $(\cdot)_+$ indicates positive rectification. To avoid the singular endpoint at $t = 1$, we truncate the path at $\tau := 1 - \varepsilon_t$ and use the exact conditional rates above for $t \in [0, \tau]$. The error analysis in Appendix A.4 shows that this truncation contributes an $O(\varepsilon_t)$ error in total variation and an $O(\sqrt{\varepsilon_t})$ error in W_2, with the bounds depending on the source–target displacement induced by endpoint coupling π. See Appendix A.1 for the detailed rate derivation.

The associated marginal transition rates (λ_t, μ_t) are obtained by averaging the conditional rates over endpoint pairs given the current state. By Proposition 1 of Holderrieth et al. [2025], they generate the marginal probability path $p_t(x) = \mathbb{E}_{(x_0,x_1)\sim\pi}[p_t(x \mid x_0, x_1)]$. By Proposition 2 of Holderrieth et al. [2025], training with the conditional rates gives the same population gradients as training with the marginal rates. Thus, count-FM uses the same conditional-to-marginal matching principle as continuous flow matching, with transition rates replacing velocities. The CTMC formulation follows prior work [Campbell et al., 2022, 2024], while the signed binomial bridge and its closed-form rates are specific to count-FM.

To ensure zero death rate in any coordinate whose count is zero, we parameterize the model by nonnegative birth rates $\lambda_\theta(x, t)$ and death rates $\mu_\theta(x, t) = x \odot \beta_\theta(x, t)$, where the death coefficients $\beta_\theta(x, t)$ are nonnegative. In the following experiments, birth and death rates are modeled using a single neural network with separate outputs. We derive the training objective from the path-space Kullback–Leibler (KL) divergence induced by the conditional count bridge, following the standard path-space relative-entropy formulation for CTMCs [Arampatzis et al., 2015]. Let $X_{(0,\tau]} := \{X_s :$

$0 < s \leq \tau\}$ and define

$$\mathcal{L}_{\text{path}}(\theta) = \mathbb{E}_{(x_0,x_1)\sim\pi}\left[\mathrm{KL}\big(p(X_{(0,\tau]} \mid X_0 = x_0, X_1 = x_1) \,\big\|\, p_\theta(X_{(0,\tau]} \mid X_0 = x_0)\big)\right]. \quad (3)$$

For the local birth-death process, the path-space KL decomposes into local one-step KL terms. Taking the infinitesimal limit and dropping terms independent of θ gives the practical training objective

$$\mathcal{L}_{\text{train}}(\theta) = \mathbb{E}_{\substack{(x_0,x_1)\sim\pi,\\ t\sim\text{Unif}[0,\tau],\\ x\sim p_t(\cdot|x_0,x_1)}}\left[\sum_{i=1}^{d} \ell(\lambda_{t,i}(x \mid x_0,x_1), \lambda_{\theta,i}(x,t)) + \sum_{i=1}^{d} \ell(\mu_{t,i}(x \mid x_0,x_1), \mu_{\theta,i}(x,t))\right], \quad (4)$$

where $\ell(u,v) = v - u\log v$. We use $\log(v+\varepsilon_\ell)$ for numerical stability. Appendices A.2 and A.3 show that $\mathcal{L}_{\text{path}}(\theta) = \tau\mathcal{L}_{\text{train}}(\theta) + c$, where c does not depend on θ. Thus the path-space and practical objectives are equivalent for optimization. This is analogous to the data-augmentation view of diffusion objectives in Kingma and Gao [2023]. Appendix A.4 further relates the rate-matching error to the final distribution error.

2.2 Sample generation

After training, we generate samples by simulating the learned birth-death process forward from $t = 0$ to $t = \tau = 1 - \varepsilon_t$. The initial state is sampled from the source distribution p_0, which can be either a simple count-valued distribution (for unconditional generation) or the observed source population. Furthermore, we use a first-order local-jump discretization. Like tau-leaping methods for Markov jump processes [Gillespie, 2001], the sampler freezes the transition rates within each time step, while restricting each coordinate to at most one jump. With K uniform steps and step size $\Delta = \tau/K = (1-\varepsilon_t)/K$, define the total jump rate for coordinate i at state x and time t as

$$r_i(x,t) = \lambda_{\theta,i}(x,t) + \mu_{\theta,i}(x,t).$$

For each coordinate i with $r_i(x,t) > 0$, we sample one of three outcomes,

$$p_i^{\text{stay}} = \exp(-r_i\Delta), \qquad p_i^{\text{birth}} = \big(1-\exp(-r_i\Delta)\big)\frac{\lambda_{\theta,i}}{r_i}, \qquad p_i^{\text{death}} = \big(1-\exp(-r_i\Delta)\big)\frac{\mu_{\theta,i}}{r_i}.$$

When $r_i(x,t) = 0$, we set $p_i^{\text{stay}} = 1$ and $p_i^{\text{birth}} = p_i^{\text{death}} = 0$. The next state is obtained by applying the sampled unit update in each coordinate. Repeating this update over the K time steps from $t = 0$ to $t = \tau$ yields an approximate sample trajectory.

Appendix A.4 decomposes the terminal distribution error into 1) endpoint truncation, 2) rate estimation, and 3) sampling discretization. The truncation bound favors a smaller endpoint gap (smaller ε_t) and couplings with shorter source–target displacements. The squared total variation error due to rate estimation is at most half the gap between the optimal and training losses. The discretization bound further differentiates between errors due to multiple jumps within each step and from rate freezing, favoring lower jump intensities ($\sum_i^d r_i(x,t)$) and rates that vary slowly in time and state. Under bounded and Lipschitz rates, the local discretization error is $O(\Delta^2)$, and the global error is $O(K^{-1})$. Our implementation, the signed binomial bridge, partly follows this guidance. Its monotone conditional trajectories minimize the number of unit jumps between fixed endpoints, with constant expected jump intensity equal to the source-target L_1 displacement. The OT coupling in Section 2.3 can also lower the expected jump intensity if it reduces the mean source-target L_1 displacement relative to independent coupling.

2.3 Endpoint coupling

The endpoint coupling $\pi(x_0,x_1)$ determines how source and target samples are paired during training and thus influences the learned marginal transition path. Here, we either consider independent coupling, $\pi_{\text{ind}}(x_0,x_1) = p_0(x_0)p_1(x_1)$, (**count-FM**) or a minibatch optimal-transport (OT) coupling (**count-FM-OT**), similar to OT-CFM [Tong et al., 2024]. For the latter, we draw source and target minibatches $\{x_{0,b}\}_{b=1}^B$ and $\{x_{1,b}\}_{b=1}^B$. These define empirical measures $\hat{p}_0^B = \frac{1}{B}\sum_{b=1}^B \delta_{x_{0,b}}$ and $\hat{p}_1^B = \frac{1}{B}\sum_{b=1}^B \delta_{x_{1,b}}$, both with uniform weights $u_B = \frac{1}{B}\mathbf{1}_B$. We then compute the empirical OT coupling

$$\Gamma^\star = \arg\min_{\Gamma\in\Pi(u_B,u_B)} \sum_{b=1}^{B}\sum_{b'=1}^{B} \Gamma_{bb'}\, c(x_{0,b}, x_{1,b'}),$$

where $\Pi(u_B, u_B)$ denotes the set of joint distributions on the two minibatches with marginals u_B. We sample endpoint pairs (x_0, x_1) from $\Gamma^\star$ and then use the same conditional bridge construction and training objective as in Section 2.1. This minibatch OT step can be viewed as an empirical approximation to a population coupling between p_0 and p_1. We use the symmetric Poisson cost

$$c(x, y) = \sum_{i=1}^{d} \left[x_i \log \frac{x_i + \varepsilon_c}{y_i + \varepsilon_c} + y_i \log \frac{y_i + \varepsilon_c}{x_i + \varepsilon_c} \right],$$

where $\varepsilon_c > 0$ is a small constant for numerical stability. This is a numerically stabilized symmetrized divergence between count vectors.

In the experiments, we report both count-FM and count-FM-OT for unconditional generation. The OT coupling changes the endpoint coupling, and hence modifies the geometry of the learned marginal transport path. In practice, OT coupling tends to induce transitions with lower curvature, which is beneficial for interpretation. It can also improve sampling efficiency, in the sense that under the same discretization scheme, similar sample quality may be achieved with fewer function evaluations (NFEs) or less wall-clock time. We verify this empirically in Appendix Figure 5, where the OT-coupled version generally reaches a given quality level more quickly than the independently coupled version.

2.4 Conditional generation

For conditional generation, we augment the model with covariates y and use conditional rates $\lambda_\theta(x, t, y)$ and $\mu_\theta(x, t, y)$. We adapt classifier-free guidance (CFG) [Ho and Salimans, 2021] to the transition rates. During training, conditioning variables are randomly dropped, allowing a single network to learn both conditional and unconditional rates.. This avoids training a separate auxiliary classifier and provides a simple way to control the strength of conditioning at sampling time through the guidance scale. At sampling time, we combine the conditional and unconditional rates as

$$\lambda_\theta^{\text{cfg}} = \left[\lambda_\theta^{\text{uncond}} + w\big(\lambda_\theta^{\text{cond}} - \lambda_\theta^{\text{uncond}}\big) \right]_+, \qquad \mu_\theta^{\text{cfg}} = \left[\mu_\theta^{\text{uncond}} + w\big(\mu_\theta^{\text{cond}} - \mu_\theta^{\text{uncond}}\big) \right]_+,$$

where $w \geq 0$ is the guidance scale and $[\cdot]_+$ ensures nonnegative rates. The operator $[\cdot]_+$ has no effect for $0 \leq w \leq 1$, but may alter the extrapolated rates when $w > 1$. Here, $w = 0$ recovers unconditional generation, $0 < w < 1$ interpolates between unconditional and conditional generation, and $w = 1$ gives the standard conditional model. Values $w > 1$ heuristically extrapolate beyond the standard conditional model, which can improve condition alignment and sharpen condition-specific structure. At the same time, overly large w may distort the marginal distribution and reduce diversity.

3 Simulation

We begin with a two-dimensional example simulation designed to evaluate both sample quality and intermediate transport behavior. The target distribution is an equal-weight mixture of two Gamma-Poisson components with modes near $(60, 5)$ and $(60, 40)$. For a fair comparison, all models except Poisson-JUMP [Chen and Zhou, 2023] use the same discrete-uniform source distribution on a square count grid covering the displayed data range, while Poisson-JUMP uses its own Poisson-based source construction. We compare both count-FM and count-FM-OT against several representative discrete generative baselines: Dirichlet-FM [Stark et al., 2024] and discrete-FM [Gat et al., 2024] from discrete flow matching, D3PM [Austin et al., 2021], tauLDR [Campbell et al., 2022], and SEDD [Lou et al., 2024] from discrete diffusion-style modeling, and Poisson-JUMP, a count-specific jump model. Most of these baselines are based on categorical-state parameterizations. Poisson-JUMP instead works directly with counts, but unlike count-FM it does not learn an explicit conditional bridge with local birth and death rates. For count-FM, we use a small time-varying MLP backbone with hidden width 32, with shared hidden layers and separate output channels for the birth and death rates. For the competing methods, we use standard or matched-capacity architectures.

We evaluate sample quality using the 1- and 2-Wasserstein distances, W_1 and W_2, and $\text{MMD}^2_{\text{RBF}}$, the squared maximum mean discrepancy with a Gaussian RBF kernel [Gretton et al., 2012]. For each model, we use 2,000 generated samples under a common evaluation seed and report the mean and standard deviation across target resamples in Table 1. count-FM-OT ranks first under both W_1 and W_2. SEDD obtains the lowest $\text{MMD}^2_{\text{RBF}}$. Thus, the main conclusion is robust to the choice of

Wasserstein order, while both count-FM variants use substantially fewer trainable parameters than the competing models.

Figure 3 in Appendix B shows representative intermediate samples along each model's native sampling trajectory. Figure 4 further compares the marginal bridges under a common progress variable, with Poisson-JUMP excluded because no analogous bridge is available. Under this normalization, both count-FM variants evolve smoothly in count space, while count-FM-OT follows a visibly straighter transition path. In contrast, the categorical-state diffusion- and flow-based baselines move most mass toward the target early along the common progress scale, indicating a more abrupt transition in count space. This geometric advantage is also reflected in sampling efficiency. Appendix Figure 5 shows that count-FM-OT generally reaches a given W_2 or $\mathrm{MMD}^2_{\mathrm{RBF}}$ level with fewer NFEs or less runtime than the independently coupled version.

Table 1: **Performance comparison on the two-dimensional simulation.** For each model, 2,000 samples are generated under a common evaluation seed, and results are reported as mean $\pm$ standard deviation across target resamples. All models except Poisson-JUMP use the same discrete-uniform source distribution. Model names include the number of trainable parameters in parentheses. Best and second-best results are bolded and underlined, respectively.

Model (params)	$W_1 \downarrow$	$W_2 \downarrow$	$\mathrm{MMD}^2_{\mathrm{RBF}} \downarrow$
count-FM-OT (2,372)	**1.685 ± 0.163**	**2.341 ± 0.330**	<u>0.0005 ± 0.0004</u>
count-FM (2,372)	1.866 ± 0.219	<u>2.583 ± 0.378</u>	0.0009 ± 0.0004
discrete-FM (14,250)	1.942 ± 0.205	2.684 ± 0.363	0.0007 ± 0.0005
D3PM (15,306)	1.982 ± 0.190	2.812 ± 0.446	0.0010 ± 0.0006
SEDD (15,306)	<u>1.747 ± 0.115</u>	3.051 ± 0.165	**0.0002 ± 0.0003**
Poisson-JUMP (67,330)	2.974 ± 0.271	3.794 ± 0.394	0.0035 ± 0.0009
tauLDR (15,306)	2.417 ± 0.292	3.855 ± 0.586	0.0024 ± 0.0012
Dirichlet-FM (15,306)	3.569 ± 0.379	6.101 ± 0.450	0.0035 ± 0.0014

4 Applications

4.1 Single-cell RNA-seq generation and transport

We study the Dentate Gyrus scRNA-seq dataset [Hochgerner et al., 2018], which contains 2,930 cells and 13,913 genes, together with cell-type and developmental-age annotations spanning multiple lineages. The two developmental time points used in our transport experiment are postnatal day 12 (P12, 1,124 cells) and postnatal day 35 (P35, 1,806 cells). We consider two tasks: unconditional generation from the marginal distribution and transport from P12 to P35 along development.

4.1.1 Unconditional generation

For unconditional generation, each model is trained to generate full gene-expression count vectors from the marginal cell distribution. We use a random 80/20 train-test split over all 2,930 cells, giving 2,344 training cells and 586 held-out test cells. We compare count-FM and count-FM-OT with representative baselines for single-cell generation, including scVI [Lopez et al., 2018] as a latent-variable baseline, scDiffusion [Luo et al., 2024] and scLDM [Palla et al., 2025] as latent diffusion models, and DCM [Bhattacharya et al., 2026] and CFGen [Palma et al., 2025] as additional recent generative baselines for scRNA count data. For both scRNA experiments, count-FM-(OT) uses a shared transformer backbone with hidden dimension 256, depth 8, and 8 attention heads, with separate output channels for the birth and death rates. The competing methods use their implementations with comparable size settings for a fair comparison.

We evaluate held-out sample quality using W_2 and $\mathrm{MMD}^2_{\mathrm{RBF}}$ computed in a PCA feature space built from the top 2,000 variable genes after normalization and log transformation, with the number of PCs chosen to explain 90% of the variance. Here, W_2 is computed between Gaussian fits to the real and generated PCA features. The results are summarized in Table 2. Count-FM achieves the best performance on both metrics, while count-FM-OT performs comparably. As in the simulation experiment (Section 3), OT endpoint coupling improves sampling efficiency at lower budgets, with count-FM-OT reaching comparable quality using fewer NFEs or less runtime, while the gap becomes

smaller at larger budgets (Appendix Figure 5). Paired comparisons across the five matched seeds show that count-FM improves both metrics over all five competing methods after Holm correction [Holm, 1979], while an additional marker-gene analysis shows that count-FM achieves the lowest marker-expression MAE, indicating the most accurate recovery of cell-type-relevant expression levels (Appendix Tables 4 and 5). Count-FM uses 45.70% fewer trainable parameters than scVI and 5.46% fewer than scLDM.

Table 2: **Performance comparison across generative models on the Dentate Gyrus dataset for unconditional generation.** Results are reported as mean ± standard deviation over 5 repeated runs. Model names include the number of trainable parameters in parentheses when available. count-FM achieves the best performance on both W_2 and $\mathrm{MMD}^2_{\mathrm{RBF}}$, while count-FM-OT performs comparably.

Model (params)	$W_2 \downarrow$	$\mathrm{MMD}^2_{\mathrm{RBF}} \downarrow$
count-FM (9,857,792)	**20.456 ± 0.055**	**0.0185 ± 0.0005**
count-FM-OT (9,857,792)	20.553 ± 0.054	0.0195 ± 0.0007
scLDM (10,426,955)	20.739 ± 0.018	0.0202 ± 0.0003
scDiffusion (34,704,897)	20.545 ± 0.051	0.0225 ± 0.0004
scVI (18,155,426)	20.687 ± 0.042	0.0229 ± 0.0002
DCM (39,848,665)	23.566 ± 0.010	0.0296 ± 0.0000
CFGen (29,165,380)	22.226 ± 0.106	0.0361 ± 0.0008

4.1.2 Transport from P12 to P35

We next study transport between postnatal days 12 (P12) and 35 (P35) in the Dentate Gyrus dataset. A key advantage of count-FM is that it defines transport directly in count space, with local ± 1 transitions that preserve the ordered count geometry and make intermediate changes gradual and interpretable. This is especially useful in developmental settings, where one would like to inspect not only the endpoints but also the full transition path. Related single-cell methods have likewise emphasized reconstructing developmental trajectories and fate structure, for example through optimal transport and fate-mapping frameworks [Schiebinger et al., 2019, Lange et al., 2022].

The Dentate Gyrus data have a dominant developmental trajectory along the granule-cell lineage, while several mature populations form comparatively stable side lineages. In particular, prior analyses [Cui et al., 2024] of this dataset identify the main progression from neuroblast cells to granule immature cells and then to granule mature cells, whereas mature side populations evolve separately rather than along the main granule trajectory [Hochgerner et al., 2018]. To avoid source-target pairings driven only by geometric proximity, we transport P12 count profiles to P35 count profiles using lineage-restricted OT couplings. Specifically, OT matching is performed only within lineage-consistent source-target groups, so the learned transport respects known developmental structure and avoids biologically implausible transitions across incompatible cell states. This restriction encourages straighter, lower-curvature paths that are easier to interpret biologically. We use separate 80/20 train-test splits for each subset, stratified by cell cluster to preserve cluster composition.

Figure 1 summarizes the P12-to-P35 transport results. In panel A, generated trajectories move smoothly from the P12 cells toward the P35 cells, with colors indicating cell lineage. In panel B, the inferred terminal states are largely lineage-consistent. Neuroblast and granule immature cells predominantly map to granule immature or granule mature states, while mature side populations such as astrocytes, endothelial, and GABA cells mostly remain within their own lineages. Because lineage-restricted OT incorporates known lineage information, this lineage consistency is partly induced by the coupling rather than discovered by the model. Appendix Figures 6 and 7 further show that the transition unfolds progressively over time on both the training and test sets. Together, these results illustrate how count-FM can be used to model structured developmental transport directly in count space.

4.2 Conditional generation on neural spike counts

We next study conditional generation of neural spike counts from a large collection of multi-region hippocampal and entorhinal cortex recordings in behaving Long-Evans rats. In our experiment, we

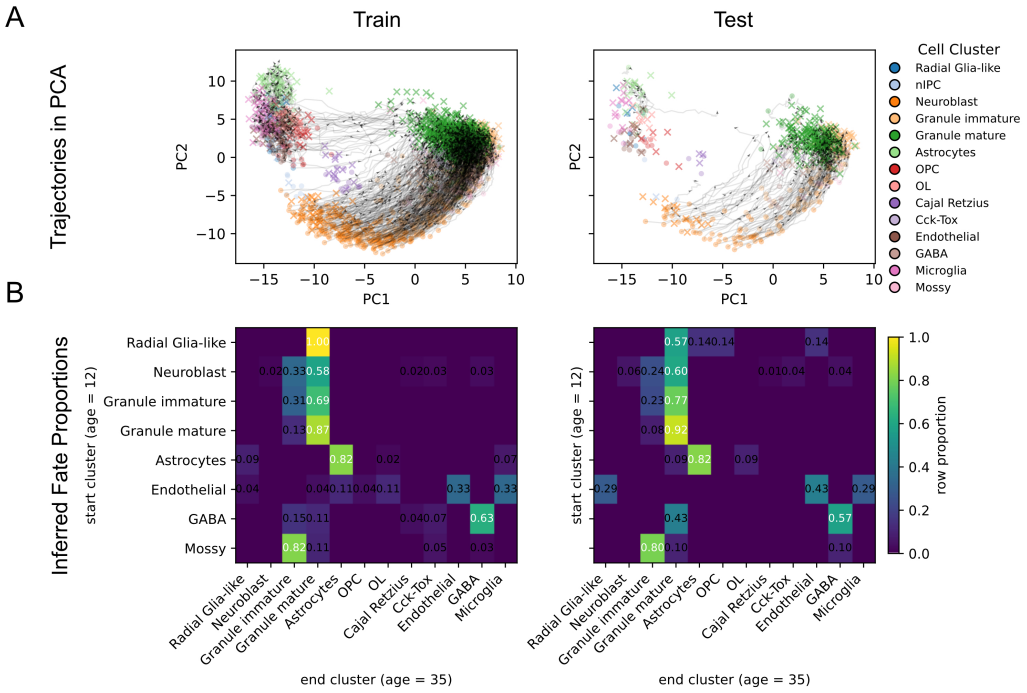

Figure 1: **Transport from P12 to P35. A.** Generated transport trajectories in PCA space for the training and test splits. Circles denote the P12 source cells and crosses denote the P35 target cells, with colors indicating cell cluster identity. Black curves show representative generated trajectories, which smoothly connect P12 to P35. **B.** Inferred fate proportions under P12-to-P35 transport. Rows correspond to source clusters at P12 and columns to inferred endpoint clusters at P35. Entries are row-normalized proportions for the training and test splits.

focus on one linear-track session (`ec013.719`) from the CRCNS hc-3 dataset [Mizuseki et al., 2013, 2014], retaining 62 simultaneously recorded units from CA1, EC2, EC3, and EC5, yielding 29,239 observations after preprocessing. We use an 80/20 train-test split, giving 23,391 training observations and 5,848 held-out observations. The covariates are linearized position and running direction, combined into a single signed position (Figure 2A). Here, in contrast to generative benchmarking (simulation, Section 3) and unconditional generation (scRNA-seq, Section 4.1.1), we focus on conditional count modeling versus standard regression baselines. Specifically, we compare count-FM with guidance scales $w \in \{0, 1, 2\}$ against two reference models, a deterministic MLP regressor, which predicts only conditional means, and a Poisson MLP baseline, which models count noise but has limited ability to capture complex correlation structure. Count-FM instead models a complicated joint conditional count distribution and supports classifier-free guidance (CFG), which controls the strength of conditional information without training an auxiliary classifier, as described in Section 2.4. For a fair comparison, all models use comparable width-128 three-layer MLP backbones with SELU activations. The MLP and Poisson MLP therefore serve as diagnostic baselines for conditional mean and covariance rather than a comprehensive generative benchmark.

Figure 2 shows representative results for location-modulated neurons from these regions. Many recorded neurons in hippocampal and entorhinal regions are silent or only weakly location-specific, including many interneuron-like units [Thompson and Best, 1989, Epsztein et al., 2011, Hangya et al., 2010], so we display a subset of neurons from each region to make the spatial response pattern visible. In the mean-response heatmaps (Figure 2A and Appendix Figure 8), all models recover the main spatial response patterns. These response patterns are well known in classic findings that hippocampal neurons encode position and can be modulated by running direction [McNaughton et al., 1983, Markus et al., 1995]. However, the models differ more clearly in their population dependence structure (Figure 2B). Count-FM with $w = 1$ better preserves the observed cross-neuron correlation pattern, whereas the Poisson MLP underestimates the dependence structure.

We further examine the effect of guidance scale in Appendix D.2. For mean responses (Appendix Figure 8), count-FM with $w = 0$ lacks conditional information and only captures marginal firing-rate differences, while $w = 2$ sharpens location-specific responses but reduces calibration to the true data. For population correlation structure (Appendix Figure 9), the MLP mean regressor is excluded because it is not a generative count model. Count-FM with $w = 0$ still captures some marginal population dependence, Poisson MLP substantially underestimates correlations, and $w = 2$ amplifies dependence at the cost of calibration, consistent with the discussion in Section 2.4.

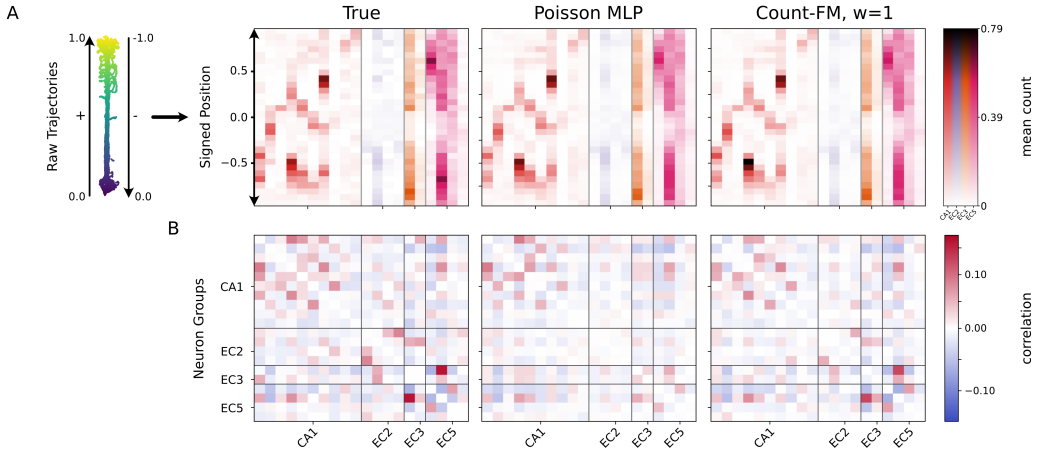

Figure 2: **Conditional generation on the hc-3 linear-track session.** Summaries are estimated from 100 generated samples per held-out covariate. **A.** Signed position is constructed from linearized position and running direction. Mean-response heatmaps show bin-wise mean counts for the true held-out data, Poisson MLP, and count-FM with $w = 1$. **B.** Population correlation matrices for the same active-neuron set. Count-FM better preserves cross-neuron dependence, while Poisson MLP underestimates population correlations.

To quantify these qualitative observations, we report several held-out metrics in Table 3. For a signed-position bin b and neuron j, let μ_{bj} denote the held-out mean count and $\hat{\mu}_{bj}$ the model mean, estimated from 50 generated samples for generative models. We evaluate $\mathrm{RMSE}_\mu = \sqrt{\frac{\sum_b n_b \sum_j (\hat{\mu}_{bj} - \mu_{bj})^2}{\sum_b n_b d}}$ and analogously compute $\mathrm{RMSE}_{\mathrm{var}}$ and RMSE_0 using the bin-wise variance and zero fraction. To evaluate correlation structure, we compute the off-diagonal covariance matrix C_b of active neurons within each bin and calculate $\mathrm{Cov}_{\mathrm{F}} = \frac{\sum_b n_b \|\hat{C}_b - C_b\|_F}{\sum_b n_b}$, where $\hat{C}_b$ is the corresponding off-diagonal covariance matrix computed from generated samples in bin b. We also summarize place-field sharpness by the raw contrast $\mathrm{contrast}_j = \frac{\max_b \hat{\mu}_{bj} - \mathrm{median}_b(\hat{\mu}_{bj})}{\mathrm{mean}_b(\hat{\mu}_{bj})}$, averaged over active neurons. Larger contrast indicates stronger peak-to-background tuning.

Table 3 summarizes results over 5 replicated runs, each using a different random seed for train-test splitting and model retraining. Count-FM with $w = 1$ gives the strongest overall generative performance, with the lowest RMSE_μ, $\mathrm{RMSE}_{\mathrm{var}}$, RMSE_0, and $\mathrm{Cov}_{\mathrm{F}}$, indicating best fit to mean firing rate, sparsity, and correlation structure. The Poisson MLP baseline remains competitive for the mean response, but its variance, zero-fraction, and covariance errors are substantially larger. The unconditional model ($w = 0$) has no spatial specificity. Increasing guidance to $w = 2$ produces the sharpest place fields, but worsens most fidelity metrics, indicating over-sharpening. Overall, $w = 1$ gives the best balance between spatial tuning and distributional calibration.

In a separate dataset, we also evaluate conditional generation on piriform cortex odor-response spike trains in Appendix D.3, where count-FM shows a similar guidance tradeoff. The mean MLP and Poisson MLP baselines recover only coarse response structure, with overly smooth or diffuse responses. In contrast, count-FM better captures odor- and respiration-dependent response patterns. Moderate guidance (e.g., $w = 1$) gives better calibration, while stronger guidance (e.g., $w = 2$) sharpens the pattern but can over-amplify spike counts.

Table 3: **Conditional generation on the hc-3 linear-track session.** Results are mean ± standard deviation across 5 random seeds. count-FM with $w = 1$ gives the best performance on mean firing rate (RMSE_μ), sparsity (RMSE_0), and covariance structure (RMSE_{var} and Cov_{F}), while $w = 2$ gives the largest contrast but worsens calibration.

Model	$\text{RMSE}_\mu \downarrow$	$\text{RMSE}_{\text{var}} \downarrow$	$\text{RMSE}_0 \downarrow$	$\text{Cov}_{\text{F}} \downarrow$	Contrast
MLP mean	0.026 ± 0.002	–	–	–	3.087 ± 0.034
Poisson MLP	0.027 ± 0.002	0.115 ± 0.003	0.027 ± 0.001	0.564 ± 0.012	3.134 ± 0.102
count-FM, $w = 0$	0.070 ± 0.002	0.110 ± 0.005	0.043 ± 0.000	0.451 ± 0.021	0.109 ± 0.007
count-FM, $w = 1$	$\mathbf{0.026 \pm 0.002}$	$\mathbf{0.048 \pm 0.005}$	$\mathbf{0.016 \pm 0.001}$	$\mathbf{0.344 \pm 0.017}$	3.209 ± 0.086
count-FM, $w = 2$	0.061 ± 0.002	0.094 ± 0.007	0.036 ± 0.001	0.451 ± 0.020	$\mathbf{4.521 \pm 0.076}$

5 Discussion

In this work, we introduced count-FM, a flow-matching framework for count data based on local birth-death dynamics. By modeling transport through local births and deaths directly in count space, count-FM respects the underlying count geometry while avoiding the large categorical-state output parameterizations used by many existing discrete generative models. In simulation and in applications to scRNA-seq and neural spike trains, count-FM achieved high sample quality with fewer parameters, while its intermediate paths remain in count space and evolve through local changes, supporting their interpretation as source-to-target transformations.

Although count-FM performed well across our experiments, the model still has several limitations and natural directions for improvement. First, the conditional binomial bridge is deliberately simple, but it may not be stochastic enough to adequately explore plausible intermediate paths. This may increase extrapolation error and make the learned source-to-target transition overly rigid. When different futures are indistinguishable from the current state, marginal rate matching averages their conditional dynamics, and additional conditioning is needed to recover fate-specific transitions. A natural extension is to replace the binomial bridge with a more flexible stochastic bridge, for example a beta-binomial variant, or more generally a latent stochastic path construction in the spirit of stochastic interpolants [Albergo and Vanden-Eijnden, 2023, Albergo et al., 2025] and GP-CFM [Wei and Ma, 2025].

Second, our current local-jump sampler is first-order and can introduce larger discretization error when jump rates are large or rapidly varying. Although model size is independent of the count range, large source-target displacements require more sampling steps. A source distribution closer to the target count scale or OT coupling can reduce typical displacements and therefore the required number of steps. The sampler itself could also be improved using nonuniform or adaptive time grids, uniformization-based CTMC simulation [Chen and Ying, 2025], more accurate bridge-aware simulation schemes [Hobolth and Stone, 2009], and inference-time correction methods [Skreta et al., 2025]. Accelerated transition operators based on shortcut models [Frans et al., 2025] or inner-flow sampling [Holderrieth et al., 2026] may further reduce the required number of sampling steps, while flow-map learning [Boffi et al., 2025, Potaptchik et al., 2026] can bypass stepwise simulation and remove this source of discretization error.

In summary, we introduced count-FM, a flow-matching framework for count data based on local birth-death dynamics. Across simulation and applications to biomedical data, including scRNA-seq and neural spike trains, count-FM achieved strong sample quality with fewer parameters than competing categorical-state approaches, while preserving count-valued intermediate paths. These results suggest that count-FM provides an effective framework for modeling high-dimensional count data by enabling generation and transport directly in the count space.

Acknowledgments and Disclosure of Funding

This work was supported by a grant from the National Institutes of Health (RF1DA056376) to JP through the BRAIN Initiative.

References

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. URL http://jmlr.org/papers/v26/23-1605.html.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Georgios Arampatzis, Markos A. Katsoulakis, and Yannis Pantazis. Pathwise sensitivity analysis in transient regimes. In Stefan Heinz and Hakima Bessaih, editors, *Stochastic Equations for Complex Systems: Theoretical and Computational Topics*, pages 105–124. Springer, Cham, 2015. doi: 10.1007/978-3-319-18206-3_5. URL https://doi.org/10.1007/978-3-319-18206-3_5.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and Geometry of Markov Diffusion Operators*, volume 348 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2014. doi: 10.1007/978-3-319-00227-9. URL https://doi.org/10.1007/978-3-319-00227-9.

Sagnik Bhattacharya, Abhiram Gorle, Ahsan Bilal, Connor Ding, Amit Kumar Singh Yadav, and Tsachy Weissman. ItDPDM: Information-theoretic discrete poisson diffusion model. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1021014b8c5a9008fc245736a1b73d7e-Abstract-Conference.html.

Sanjukta Bhattacharya, Christian Gensbigler, Shaamil Karim, and Jon Lees. Discrete diffusion for single-cell gene expression modeling. In *ICLR 2026 Workshop on Machine Learning for Genomics Explorations*, 2026. URL https://openreview.net/forum?id=GPR1YXdE4U.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=cqDH0e6ak2.

Kevin A Bolding and Kevin M Franks. Complementary codes for odor identity and intensity in olfactory cortex. *eLife*, 6:e22630, apr 2017. ISSN 2050-084X. doi: 10.7554/eLife.22630. URL https://doi.org/10.7554/eLife.22630.

Kevin A Bolding, Shivathmihai Nagappan, Bao-Xia Han, Fan Wang, and Kevin M Franks. Recurrent circuitry is required to stabilize piriform cortex odor representations across brain states. *eLife*, 9:e53125, jul 2020. ISSN 2050-084X. doi: 10.7554/eLife.53125. URL https://doi.org/10.7554/eLife.53125.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=DmT862YAieY.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5453–5512. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/campbell24a.html.

Chen-Hao Chao, Wei-Fang Sun, Hanwen Liang, Chun-Yi Lee, and Rahul G. Krishnan. Beyond masked and unmasked: Discrete diffusion models via partial masking. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://nips.cc/virtual/2025/poster/116103.

Hongrui Chen and Lexing Ying. Convergence analysis of discrete diffusion model: Exact implementation through uniformization. *Journal of Machine Learning*, 4(2):108–127, 2025. doi: 10.4208/jml.240812. URL https://doi.org/10.4208/jml.240812.

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 4735–4763. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23q.html.

Tianqi Chen and Mingyuan Zhou. Learning to jump: Thinning and thickening latent counts for generative modeling. In *International Conference on Machine Learning (ICML)*, 2023.

Haotian Cui, Hassaan Maan, Maria C. Vladoiu, Jiao Zhang, Michael D. Taylor, and Bo Wang. Deepvelo: deep learning extends rna velocity to multi-lineage systems with cell-specific kinetics. *Genome Biology*, 25(1):27, 2024. doi: 10.1186/s13059-023-03148-9. URL https://doi.org/10.1186/s13059-023-03148-9.

Jérôme Epsztein, Michael Brecht, and Albert K. Lee. Intracellular determinants of hippocampal ca1 place and silent cell activity in a novel environment. *Neuron*, 70(1):109–120, 2011. doi: 10.1016/j.neuron.2011.03.006. URL https://doi.org/10.1016/j.neuron.2011.03.006.

Nic Fishman, Gokul Gowri, Tanush Kumar, Jiaqi Lu, Valentin de Bortoli, Jonathan S. Gootenberg, and Omar Abudayyeh. Count bridges enable modeling and deconvolving transcriptomic data. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4nOZBufbLC.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OlzB6LnXcS.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GTDKo3Sv9p.

Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001. doi: 10.1063/1.1378322. URL https://doi.org/10.1063/1.1378322.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf.

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

Balázs Hangya, Yu Li, Robert U. Muller, and András Czurkó. Complementary spatial firing in place cell–interneuron pairs. *Journal of Physiology*, 588(21):4165–4175, 2010. doi: 10.1113/jphysiol.2010.194274. URL https://doi.org/10.1113/jphysiol.2010.194274.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2722–2730. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ho19a.html.

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

Asger Hobolth and Eric A. Stone. Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution. *The Annals of Applied Statistics*, 3(3):1204 – 1231, 2009. doi: 10.1214/09-AOAS247. URL https://doi.org/10.1214/09-AOAS247.

Hannah Hochgerner, Amit Zeisel, Peter Lönnerberg, and Sten Linnarsson. Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing. *Nature Neuroscience*, 21(2):290–299, February 2018. doi: 10.1038/s41593-017-0056-2.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky T. Q. Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=RuP17cJtZo.

Peter Holderrieth, Uriel Singer, Tommi Jaakkola, Ricky T. Q. Chen, Yaron Lipman, and Brian Karrer. GLASS flows: Efficient inference for reward alignment of flow and diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vH7OAPZ2dR.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Emiel Hoogeboom, Jorn Peters, Rianne van den Berg, and Max Welling. Integer discrete flows and lossless compression. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/9e9a30b74c49d07d8150c8c83b1ccf07-Paper.pdf.

Emiel Hoogeboom, Taco Cohen, and Jakub Mikolaj Tomczak. Learning discrete distributions by dequantization. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021. URL https://openreview.net/forum?id=a0EpGhKt_R.

Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=NnMEadcdyD.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

Marius Lange, Volker Bergen, Michal Klein, Manu Setty, Bernhard Reuter, Mostafa Bakhti, Heiko Lickert, Meshal Ansari, Janine Schniering, Herbert B. Schiller, Dana Pe'er, and Fabian J. Theis. Cellrank for directed single-cell fate mapping. *Nature Methods*, 19(2):159–170, February 2022. doi: 10.1038/s41592-021-01346-6. URL https://doi.org/10.1038/s41592-021-01346-6.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, December 2018. doi: 10.1038/s41592-018-0229-2.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

Jianfeng Lu. A mathematical introduction to diffusion models, 2026. URL https://arxiv.org/abs/2607.01693.

Erpai Luo, Minsheng Hao, Lei Wei, and Xuegong Zhang. scdiffusion: conditional generation of high-quality single-cell data using diffusion model. *Bioinformatics*, 40(9):btae518, 08 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae518. URL https://doi.org/10.1093/bioinformatics/btae518.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7192–7203. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/luo21a.html.

Etan J. Markus, Y. L. Qin, B. Leonard, William E. Skaggs, Bruce L. McNaughton, and Carol A. Barnes. Interactions between location and task affect the spatial and directional firing of hippocampal neurons. *Journal of Neuroscience*, 15(11):7079–7094, 1995. doi: 10.1523/JNEUROSCI.15-11-07079.1995. URL https://www.jneurosci.org/content/15/11/7079.

B. L. McNaughton, C. A. Barnes, and J. O'Keefe. The contributions of position, direction, and velocity to single unit activity in the hippocampus of freely-moving rats. *Experimental Brain Research*, 52:41–49, 1983. doi: 10.1007/BF00237147. URL https://doi.org/10.1007/BF00237147.

Keiji Miura, Zachary F. Mainen, and Naoshige Uchida. Odor representations in olfactory cortex: Distributed rate coding and decorrelated population activity. *Neuron*, 74(6):1087–1098, June 2012. doi: 10.1016/j.neuron.2012.04.021. URL https://doi.org/10.1016/j.neuron.2012.04.021.

Kenji Mizuseki, Anton Sirota, Eva Pastalkova, Kamran Diba, and György Buzsáki. Multiple single unit recordings from different rat hippocampal and entorhinal regions while the animals were performing multiple behavioral tasks, 2013.

Kenji Mizuseki, Kamran Diba, Eva Pastalkova, Jeff Teeters, Anton Sirota, and György Buzsáki. Neurosharing: large-scale data sets (spike, lfp) recorded from the hippocampal-entorhinal system in behaving rats. *F1000Research*, 3:98, 2014. doi: 10.12688/f1000research.3895.2.

Giovanni Palla, Sudarshan Babu, Payam Dibaeinia, Donghui Li, Aly A Khan, Theofanis Karaletsos, and Jakub M. Tomczak. A scalable latent diffusion model for single-cell gene expression data. In *NeurIPS 2025 Workshop on AI Virtual Cells and Instruments: A New Era in Drug Discovery and Development*, 2025. URL https://openreview.net/forum?id=JQE2Fc1U0u.

Alessandro Palma, Till Richter, Hanyi Zhang, Manuel Lubetzki, Alexander Tong, Andrea Dittadi, and Fabian J Theis. Multi-modal and multi-attribute generation of single cells with CFGen. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3MnMGLctKb.

Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S. Albergo. Discrete flow maps, 2026. URL https://arxiv.org/abs/2604.09784.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2019.01.006. URL https://www.sciencedirect.com/science/article/pii/S009286741930039X.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Vhc0KrcqWu.

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.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=syXFAVqx85.

Lucien T. Thompson and Phillip J. Best. Place cells and silent cells in the hippocampus of freely-behaving rats. *Journal of Neuroscience*, 9(7):2382–2390, 1989. doi: 10.1523/JNEUROSCI.09-07-02382.1989. URL https://www.jneurosci.org/content/9/7/2382.

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. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009. doi: 10.1007/978-3-540-71050-9. URL https://doi.org/10.1007/978-3-540-71050-9.

Ganchao Wei and Li Ma. Stream-level flow matching with gaussian processes. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qg9p1I5lmp.

A Method details and derivations

This appendix provides derivations and technical details for the construction in Section 2. We first derive the conditional bridge rates, then derive the local rate-matching objective and its path-space KL interpretation. We finally analyze the endpoint truncation, rate-estimation, and sampling-discretization errors of count-FM.

A.1 Derivation of conditional bridge rates

This subsection derives the conditional birth and death rates used in Section 2.1. We work in one dimension and suppress the coordinate index i. Let $x_0, x_1 \in \mathbb{N}_0$ and define the bridge

$$X_t = x_0 + \operatorname{sgn}(x_1 - x_0)\, B_t, \qquad B_t \sim \operatorname{Binomial}(|x_1 - x_0|, t)\,.$$

For fixed endpoints (x_0, x_1), let

$$p_t(x \mid x_0, x_1) = \mathbb{P}(X_t = x \mid x_0, x_1)$$

denote the one-dimensional conditional bridge pmf. Then mass preservation is

$$\begin{aligned}\partial_t p_t(x \mid x_0, x_1) = {} & \lambda_t(x-1 \mid x_0, x_1)\, p_t(x-1 \mid x_0, x_1) + \mu_t(x+1 \mid x_0, x_1)\, p_t(x+1 \mid x_0, x_1) \\ & - \big(\lambda_t(x \mid x_0, x_1) + \mu_t(x \mid x_0, x_1)\big) p_t(x \mid x_0, x_1). \end{aligned} \tag{5}$$

This is the one-dimensional special case of the Kolmogorov forward equation (KFE) for CTMCs [Campbell et al., 2022].

Case 1: $x_1 \geq x_0$**.** In this case, the bridge is increasing, with

$$p_t(x \mid x_0, x_1) = \binom{x_1 - x_0}{x - x_0} t^{\,x - x_0} (1-t)^{\,x_1 - x}, \qquad x \in \{x_0, \ldots, x_1\},$$

and $\mu_t(x \mid x_0, x_1) = 0$ for $0 \leq t \leq 1$. Hence mass preservation reduces to

$$\partial_t p_t(x \mid x_0, x_1) = \lambda_t(x-1 \mid x_0, x_1)\, p_t(x-1 \mid x_0, x_1) - \lambda_t(x \mid x_0, x_1)\, p_t(x \mid x_0, x_1). \quad (6)$$

Differentiating the binomial pmf gives

$$\partial_t p_t(x \mid x_0, x_1) = \left(\frac{x - x_0}{t} - \frac{x_1 - x}{1-t} \right) p_t(x \mid x_0, x_1),$$

and

$$\frac{p_t(x-1 \mid x_0, x_1)}{p_t(x \mid x_0, x_1)} = \frac{x - x_0}{x_1 - x + 1} \cdot \frac{1-t}{t}.$$

Let

$$\lambda_t(x \mid x_0, x_1) = \frac{x_1 - x}{1-t},$$

then

$$\lambda_t(x-1 \mid x_0, x_1)\, p_t(x-1 \mid x_0, x_1) = \frac{x - x_0}{t}\, p_t(x \mid x_0, x_1),$$

and hence (6) holds. Therefore, for $x_1 \geq x_0$,

$$\lambda_t(x \mid x_0, x_1) = \frac{x_1 - x}{1-t}, \qquad \mu_t(x \mid x_0, x_1) = 0.$$

Case 2: $x_1 < x_0$**.** In this case, the bridge is decreasing, with

$$p_t(x \mid x_0, x_1) = \binom{x_0 - x_1}{x_0 - x} t^{\,x_0 - x} (1-t)^{\,x - x_1}, \qquad x \in \{x_1, \ldots, x_0\},$$

and $\lambda_t(x \mid x_0, x_1) = 0$ for $0 \leq t \leq 1$. Mass preservation reduces to

$$\partial_t p_t(x \mid x_0, x_1) = \mu_t(x+1 \mid x_0, x_1)\, p_t(x+1 \mid x_0, x_1) - \mu_t(x \mid x_0, x_1)\, p_t(x \mid x_0, x_1). \quad (7)$$

Applying the same calculation as in Case 1 to the reduced mass-preservation equation (7) gives

$$\mu_t(x \mid x_0, x_1) = \frac{x - x_1}{1-t}, \qquad \lambda_t(x \mid x_0, x_1) = 0.$$

Combining the two cases yields

$$\lambda_t(x \mid x_0, x_1) = \frac{(x_1 - x)_+}{1-t}, \qquad \mu_t(x \mid x_0, x_1) = \frac{(x - x_1)_+}{1-t}.$$

Applying this coordinatewise gives the formula in Section 2.1.

A.2 Local KL derivation of the rate-matching training objective

This subsection derives the rate-matching training objective (4) in Section 2.1 from the infinitesimal KL divergence between the target and model one-step transition probabilities.

Fix a bridge state x at time t. For a small step size $h > 0$, define the target local transition distribution $q_h(\cdot \mid x, x_0, x_1)$ by

$$q_h(x + e_i \mid x, x_0, x_1) = h\, \lambda_{t,i}(x \mid x_0, x_1) + o(h),$$
$$q_h(x - e_i \mid x, x_0, x_1) = h\, \mu_{t,i}(x \mid x_0, x_1) + o(h),$$

and

$$q_h(x \mid x, x_0, x_1) = 1 - h \sum_{i=1}^{d} \big(\lambda_{t,i}(x \mid x_0, x_1) + \mu_{t,i}(x \mid x_0, x_1) \big) + o(h).$$

The model for local transition probability $q_h^\theta(\cdot \mid x, t)$ is defined analogously by replacing $\lambda_{t,i}(x \mid x_0, x_1), \mu_{t,i}(x \mid x_0, x_1)$ with $\lambda_{\theta,i}(x,t), \mu_{\theta,i}(x,t)$.

Let

$$a_t(x \mid x_0, x_1) = \sum_{i=1}^{d} \big(\lambda_{t,i}(x \mid x_0, x_1) + \mu_{t,i}(x \mid x_0, x_1)\big), \qquad a_t^\theta(x,t) = \sum_{i=1}^{d} \big(\lambda_{\theta,i}(x,t) + \mu_{\theta,i}(x,t)\big).$$

We consider the local KL divergence

$$\mathrm{KL}\big(q_h(\cdot \mid x, x_0, x_1) \,\big\|\, q_h^\theta(\cdot \mid x, t)\big)\,.$$

For the no-jump event,

$$q_h(x \mid x, x_0, x_1) \log \frac{q_h(x \mid x, x_0, x_1)}{q_h^\theta(x \mid x, t)} = \big(1 - h\, a_t(x \mid x_0, x_1) + o(h)\big) \log \frac{1 - h\, a_t(x \mid x_0, x_1) + o(h)}{1 - h\, a_t^\theta(x,t) + o(h)}.$$

As $h \to 0$, using $\log(1+u) = u + o(u)$ as $u \to 0$, and noting that $q_h(x \mid x, x_0, x_1) = 1 + O(h)$,

$$q_h(x \mid x, x_0, x_1) \log \frac{q_h(x \mid x, x_0, x_1)}{q_h^\theta(x \mid x, t)} = h\big(a_t^\theta(x,t) - a_t(x \mid x_0, x_1)\big) + o(h).$$

For the jump events,

$$q_h(x + e_i \mid x, x_0, x_1) \log \frac{q_h(x + e_i \mid x, x_0, x_1)}{q_h^\theta(x + e_i \mid x, t)} = h\, \lambda_{t,i}(x \mid x_0, x_1) \log \frac{\lambda_{t,i}(x \mid x_0, x_1)}{\lambda_{\theta,i}(x,t)} + o(h),$$

and similarly,

$$q_h(x - e_i \mid x, x_0, x_1) \log \frac{q_h(x - e_i \mid x, x_0, x_1)}{q_h^\theta(x - e_i \mid x, t)} = h\, \mu_{t,i}(x \mid x_0, x_1) \log \frac{\mu_{t,i}(x \mid x_0, x_1)}{\mu_{\theta,i}(x,t)} + o(h).$$

Summing all contributions gives

$$\begin{aligned}
\mathrm{KL}\big(q_h(\cdot \mid x, x_0, x_1) \,\big\|\, q_h^\theta(\cdot \mid x, t)\big) = h \sum_{i=1}^{d} \Bigg[& \lambda_{t,i}(x \mid x_0, x_1) \log \frac{\lambda_{t,i}(x \mid x_0, x_1)}{\lambda_{\theta,i}(x,t)} \\
& - \lambda_{t,i}(x \mid x_0, x_1) + \lambda_{\theta,i}(x,t) \Bigg] \\
+ h \sum_{i=1}^{d} \Bigg[& \mu_{t,i}(x \mid x_0, x_1) \log \frac{\mu_{t,i}(x \mid x_0, x_1)}{\mu_{\theta,i}(x,t)} \\
& - \mu_{t,i}(x \mid x_0, x_1) + \mu_{\theta,i}(x,t) \Bigg] + o(h).
\end{aligned}$$

Dividing by h and letting $h \to 0$ yields

$$\begin{aligned}
\lim_{h \to 0} \frac{1}{h} \mathrm{KL}\big(q_h(\cdot \mid x, x_0, x_1) \,\big\|\, q_h^\theta(\cdot \mid x, t)\big) = & \sum_{i=1}^{d} D_{\mathrm{GKL}}(\lambda_{t,i}(x \mid x_0, x_1), \lambda_{\theta,i}(x,t)) \\
& + \sum_{i=1}^{d} D_{\mathrm{GKL}}(\mu_{t,i}(x \mid x_0, x_1), \mu_{\theta,i}(x,t))\,,
\end{aligned}$$

where D_{GKL} denotes the generalized KL divergence

$$D_{\mathrm{GKL}}(u, v) = u \log \frac{u}{v} - u + v.$$

Up to terms independent of θ, this is exactly

$$\sum_{i=1}^{d} \Big[\lambda_{\theta,i}(x,t) - \lambda_{t,i}(x \mid x_0, x_1) \log \lambda_{\theta,i}(x,t)\Big] + \sum_{i=1}^{d} \Big[\mu_{\theta,i}(x,t) - \mu_{t,i}(x \mid x_0, x_1) \log \mu_{\theta,i}(x,t)\Big].$$

Taking expectations over (x_0, x_1), $t \sim \mathrm{Unif}[0, \tau]$, and x gives the rate-matching objective in (4), up to an additive constant independent of θ.

A.3 Path-space KL interpretation of the training objective

This subsection integrates the local KL calculation in Appendix A.2 over time to obtain the path-space KL objective in (3).

Let $X_{(t,\tau]} := \{X_s : t < s \le \tau\}$ and define

$$K_\theta(t) := \mathbb{E}_{\substack{(x_0,x_1)\sim\pi,\\ x\sim p_t(\cdot|x_0,x_1)}} \left[\mathrm{KL}\big(p(X_{(t,\tau]} \mid X_t = x, x_0, x_1) \,\big\|\, p_\theta(X_{(t,\tau]} \mid X_t = x)\big)\right].$$

For $0 < h \le \tau - t$, the Markov property gives

$$p(X_{(t,\tau]} \mid X_t = x, x_0, x_1) = p(X_{(t,t+h]} \mid X_t = x, x_0, x_1)\, p(X_{(t+h,\tau]} \mid X_{t+h}, x_0, x_1),$$
$$p_\theta(X_{(t,\tau]} \mid X_t = x) = p_\theta(X_{(t,t+h]} \mid X_t = x)\, p_\theta(X_{(t+h,\tau]} \mid X_{t+h}).$$

Hence, by the chain rule for KL,

$$\begin{aligned}
&\mathrm{KL}\big(p(X_{(t,\tau]} \mid X_t = x, x_0, x_1) \,\big\|\, p_\theta(X_{(t,\tau]} \mid X_t = x)\big) \\
&= \mathrm{KL}\big(p(X_{(t,t+h]} \mid X_t = x, x_0, x_1) \,\big\|\, p_\theta(X_{(t,t+h]} \mid X_t = x)\big) \\
&\quad + \mathbb{E}_{x'\sim q_h(\cdot|x,x_0,x_1)} \left[\mathrm{KL}\big(p(X_{(t+h,\tau]} \mid X_{t+h} = x', x_0, x_1) \,\big\|\, p_\theta(X_{(t+h,\tau]} \mid X_{t+h} = x')\big)\right].
\end{aligned}$$

After averaging over $(x_0, x_1) \sim \pi$ and $x \sim p_t(\cdot \mid x_0, x_1)$, the second term is $K_\theta(t+h)$. Therefore,

$$K_\theta(t) - K_\theta(t+h) = \mathbb{E}_{\substack{(x_0,x_1)\sim\pi,\\ x\sim p_t(\cdot|x_0,x_1)}} \left[\mathrm{KL}\big(p(X_{(t,t+h]} \mid X_t = x, x_0, x_1) \,\big\|\, p_\theta(X_{(t,t+h]} \mid X_t = x)\big)\right].$$

Under standard CTMC regularity conditions, the additional path information within $(t, t+h]$ contributes only $o(h)$ beyond the endpoint transition. Hence,

$$\begin{aligned}
&\mathrm{KL}\big(p(X_{(t,t+h]} \mid X_t = x, x_0, x_1) \,\big\|\, p_\theta(X_{(t,t+h]} \mid X_t = x)\big) \\
&\quad = \mathrm{KL}\big(q_h(\cdot \mid x, x_0, x_1) \,\big\|\, q_h^\theta(\cdot \mid x, t)\big) + o(h).
\end{aligned}$$

Consequently,

$$K_\theta(t) - K_\theta(t+h) = \mathbb{E}_{\substack{(x_0,x_1)\sim\pi,\\ x\sim p_t(\cdot|x_0,x_1)}} \left[\mathrm{KL}\big(q_h(\cdot \mid x, x_0, x_1) \,\big\|\, q_h^\theta(\cdot \mid x, t)\big)\right] + o(h).$$

Dividing by h and letting $h \to 0$,

$$-\frac{d}{dt}K_\theta(t) = \mathbb{E}_{\substack{(x_0,x_1)\sim\pi,\\ x\sim p_t(\cdot|x_0,x_1)}} \left[\lim_{h\to 0} \frac{1}{h}\mathrm{KL}\big(q_h(\cdot \mid x, x_0, x_1, t) \,\big\|\, q_h^\theta(\cdot \mid x, t)\big)\right].$$

By Appendix A.2, the expected local rate-matching loss at time t equals $-\frac{d}{dt}K_\theta(t)$ up to a term independent of θ. Since $K_\theta(\tau) = 0$, integrating over $t \in [0, \tau]$ and using the uniform-time objective in (4) gives

$$K_\theta(0) = \tau \mathcal{L}_{\mathrm{train}}(\theta) + c,$$

where c is independent of θ. By definition, $K_\theta(0) = \mathcal{L}_{\mathrm{path}}(\theta)$, which gives the relation stated in Section 2.1.

A.4 Error analysis

In this section, we quantify three sources of error in count-FM, namely endpoint truncation, rate estimation, and sampling discretization. This decomposition parallels error analyses for diffusion samplers that separate endpoint, estimation, and discretization errors [Chen et al., 2023, Lu, 2026]. Let $\tau := 1 - \varepsilon_t$. Let p_t^θ denote the marginal distribution at time t of the continuous-time birth-death process with learned rates $(\lambda_\theta, \mu_\theta)$, initialized from p_0. For a sampling grid $0 = t_0 < \cdots < t_K = \tau$ with K time intervals and $\Delta_k := t_{k+1} - t_k$, let $\widehat{p}_\tau^\theta$ denote the terminal distribution produced by the discretized sampler in Section 2.2. For distributions p and q on $\mathbb{N}_0^d$, define the total variation (TV) distance as $D_{\mathrm{TV}}(p, q) := \frac{1}{2}\sum_{x\in\mathbb{N}_0^d} |p(x) - q(x)|$. By the triangle inequality,

$$D_{\mathrm{TV}}(p_1, \widehat{p}_\tau^\theta) \le \underbrace{D_{\mathrm{TV}}(p_1, p_\tau)}_{\text{endpoint truncation error}} + \underbrace{D_{\mathrm{TV}}(p_\tau, p_\tau^\theta)}_{\text{rate estimation error}} + \underbrace{D_{\mathrm{TV}}(p_\tau^\theta, \widehat{p}_\tau^\theta)}_{\text{discretization error}}.$$

We bound these three terms separately below.

A.4.1 Endpoint truncation error

For an endpoint pair $(x_0, x_1) \sim \pi$, define $n_i := |x_1^{(i)} - x_0^{(i)}|$ and $s_i := \mathrm{sgn}(x_1^{(i)} - x_0^{(i)})$. By the conditional bridge (1), $X_\tau^{(i)} = x_0^{(i)} + s_i B_\tau^{(i)}$, with $B_\tau^{(i)} \sim \mathrm{Binomial}(n_i, \tau)$. Since $x_1^{(i)} = x_0^{(i)} + s_i n_i$, the number of remaining unit jumps satisfies

$$R_i := |x_1^{(i)} - X_\tau^{(i)}| = n_i - B_\tau^{(i)} \sim \mathrm{Binomial}(n_i, \varepsilon_t).$$

Using (X_τ, X_1) from the same bridge as a coupling, note that for any coupling (X, Y) of p and q, $|p(A) - q(A)| \le \mathbb{P}(X \neq Y)$ for every event A. Taking the supremum over A gives the coupling inequality $D_{\mathrm{TV}}(p, q) \le \mathbb{P}(X \neq Y)$, and hence $D_{\mathrm{TV}}(p_\tau, p_1) \le \mathbb{P}(X_\tau \neq X_1)$. Conditionally on (x_0, x_1), reaching x_1 requires completing all n_i unit jumps in every coordinate. Since the bridge coordinates are conditionally independent,

$$\mathbb{P}(X_\tau = X_1 \mid x_0, x_1) = \prod_{i=1}^{d} \mathbb{P}(B_\tau^{(i)} = n_i) = \prod_{i=1}^{d} (1 - \varepsilon_t)^{n_i} = (1 - \varepsilon_t)^{\|x_1 - x_0\|_1}.$$

Therefore, by the law of total expectation,

$$D_{\mathrm{TV}}(p_\tau, p_1) \le \mathbb{E}_\pi \left[1 - (1 - \varepsilon_t)^{\|X_1 - X_0\|_1} \right] \le \varepsilon_t \, \mathbb{E}_\pi \|X_1 - X_0\|_1,$$

where the second inequality follows from Bernoulli's inequality, $1 - (1 - \varepsilon)^n \le n\varepsilon$. Thus the endpoint truncation error is $O(\varepsilon_t)$ in total variation.

Although the TV bound above depends on endpoint displacement through the chosen bridge coupling, total variation itself does not account for the geometric distance between count states. We therefore also consider a complementary 2-Wasserstein distance W_2 bound, which quantifies the magnitude of the remaining displacement in count space [Villani, 2009]. Recall that

$$W_2^2(p, q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(X,Y)\sim\gamma} \|X - Y\|_2^2.$$

Assume $\mathbb{E}_\pi \|X_1 - X_0\|_2^2 < \infty$. Since (X_τ, X_1) is one admissible coupling of p_τ and p_1, the definition of W_2 gives $W_2^2(p_\tau, p_1) \le \mathbb{E}\|X_\tau - X_1\|_2^2$.

Since $|X_\tau^{(i)} - X_1^{(i)}| = R_i$ and, for each coordinate i,

$$\mathbb{E}[R_i^2 \mid x_0, x_1] = n_i \varepsilon_t (1 - \varepsilon_t) + n_i^2 \varepsilon_t^2,$$

we have

$$\begin{aligned} \mathbb{E}\big[\|X_\tau - X_1\|_2^2 \mid x_0, x_1\big] &= \sum_{i=1}^{d} \mathbb{E}[R_i^2 \mid x_0, x_1] \\ &= \varepsilon_t (1 - \varepsilon_t) \|x_1 - x_0\|_1 + \varepsilon_t^2 \|x_1 - x_0\|_2^2. \end{aligned}$$

Averaging over π gives

$$W_2^2(p_\tau, p_1) \le \varepsilon_t (1 - \varepsilon_t) \mathbb{E}_\pi \|X_1 - X_0\|_1 + \varepsilon_t^2 \mathbb{E}_\pi \|X_1 - X_0\|_2^2.$$

Hence $W_2(p_\tau, p_1) = O(\sqrt{\varepsilon_t})$ for fixed endpoint moments. The quantity $\|X_1 - X_0\|_1$ is exactly the number of unit jumps required by the conditional bridge, showing how the source distribution and endpoint coupling affect truncation difficulty.

A.4.2 Rate estimation error

Let (λ_t, μ_t) denote the exact marginal transition rates introduced in Section 2.1. Define

$$\mathcal{E}_{\mathrm{rate}} = \int_0^\tau \mathbb{E}_{X_t \sim p_t} \sum_{i=1}^{d} \Big[D_{\mathrm{GKL}}(\lambda_{t,i}(X_t), \lambda_{\theta,i}(X_t, t)) + D_{\mathrm{GKL}}(\mu_{t,i}(X_t), \mu_{\theta,i}(X_t, t)) \Big] dt,$$

where D_{GKL} is defined in Appendix A.2. Let $\mathbb{P}$ and $\mathbb{P}_\theta$ denote the path distributions of the exact and learned marginal CTMCs on $[0, \tau]$. Under standard integrability and absolute-continuity conditions, the path-space KL argument [Arampatzis et al., 2015] of Appendix A.3 gives $\mathrm{KL}(\mathbb{P} \| \mathbb{P}_\theta) = \mathcal{E}_{\mathrm{rate}}$.

Since p_τ and p_τ^θ are the corresponding distributions at time τ, the data-processing inequality gives $\mathrm{KL}(p_\tau \| p_\tau^\theta) \le \mathrm{KL}(\mathbb{P} \| \mathbb{P}_\theta) = \mathcal{E}_{\mathrm{rate}}$. Pinsker's inequality [Bakry et al., 2014] therefore gives

$$D_{\mathrm{TV}}(p_\tau, p_\tau^\theta) \le \sqrt{\frac{1}{2}\mathrm{KL}(p_\tau \| p_\tau^\theta)} \le \sqrt{\frac{1}{2}\,\mathcal{E}_{\mathrm{rate}}}.$$

We next relate $\mathcal{E}_{\mathrm{rate}}$ to the training objective. Let $\mathcal{L}^*_{\mathrm{train}}$ denote the value of (4) when the model rates are replaced by the exact marginal rates (λ_t, μ_t). By definition, the marginal rate averages the conditional bridge rates over endpoint pairs given the current state, $\lambda_{t,i}(X_t) = \mathbb{E}_{(X_0,X_1)\sim\pi, X_t\sim p_t(\cdot|X_0,X_1)}[\lambda_{t,i}(X_t \mid X_0, X_1) \mid X_t]$, and analogously for the death rates. Since $\ell(u, v) = v - u \log v$ is affine in u,

$$\mathbb{E}[\ell(\lambda_{t,i}(X_t \mid X_0, X_1), v) \mid X_t] = v - \mathbb{E}[\lambda_{t,i}(X_t \mid X_0, X_1) \mid X_t] \log v = \ell(\lambda_{t,i}(X_t), v)\,.$$

Thus, after averaging over endpoint pairs, the population conditional rate-matching objective is equivalent to matching the exact marginal rates.

Recall that the rate-matching loss in (4) is $\ell(u, v) = v - u \log v$. For a marginal target rate u and model rate v, the excess loss over the oracle choice $v = u$ is

$$\ell(u, v) - \ell(u, u) = u \log \frac{u}{v} - u + v = D_{\mathrm{GKL}}(u, v).$$

Therefore, summing this excess loss over all birth and death coordinates, averaging over $X_t \sim p_t$, and integrating over $t \in [0, \tau]$ gives $\tau\,[\mathcal{L}_{\mathrm{train}}(\theta) - \mathcal{L}^*_{\mathrm{train}}] = \mathcal{E}_{\mathrm{rate}}$. Combining this identity with the TV bound above gives

$$D_{\mathrm{TV}}(p_\tau, p_\tau^\theta) \le \sqrt{\frac{\tau}{2}\,[\mathcal{L}_{\mathrm{train}}(\theta) - \mathcal{L}^*_{\mathrm{train}}]}.$$

Thus minimizing the excess rate-matching loss controls not only local transition-rate error but also the terminal distribution error of the learned continuous-time process.

A.4.3 Discretization error

For coordinate i, recall $r_i(x, t) = \lambda_{\theta,i}(x, t) + \mu_{\theta,i}(x, t)$ from Section 2.2, and define the total jump rate $a_t^\theta(x, t) = \sum_{i=1}^d r_i(x, t)$ (Appendix A.2). Also define the rate vector $\nu_\theta(x, t) := (\lambda_\theta(x, t), \mu_\theta(x, t))$. Assume $a_t^\theta(x, t) \le A$ and

$$\|\nu_\theta(y, s) - \nu_\theta(x, t)\|_1 \le L_x \|y - x\|_1 + L_t |s - t|$$

for all $x, y \in \mathbb{N}_0^d$ and $s, t \in [0, \tau]$, where L_x and L_t are the Lipschitz constants for state and time, respectively.

Let $P_k^\theta(x, \cdot)$ denote the exact learned CTMC transition over $[t_k, t_{k+1}]$ starting from x, and let $\widehat{P}_k^\theta(x, \cdot)$ denote the corresponding transition of the implemented local-jump sampler. To separate the two sources of one-step discretization error, define a frozen-rate transition over $[t_k, t_{k+1}]$. Starting from x at t_k, the rates evaluated at (x, t_k) are held constant throughout the interval. If z denotes the current state within the interval $[t_k, t_{k+1}]$,

$$\bar{\lambda}_i(z) = \lambda_{\theta,i}(x, t_k), \qquad \bar{\mu}_i(z) = \mu_{\theta,i}(x, t_k)\mathbf{1}\{z^{(i)} > 0\}.$$

Let $\bar{P}_k^\theta(x, \cdot)$ denote the resulting transition distribution. Then

$$D_{\mathrm{TV}}\left(P_k^\theta(x, \cdot), \widehat{P}_k^\theta(x, \cdot)\right) \le \underbrace{D_{\mathrm{TV}}\left(P_k^\theta(x, \cdot), \bar{P}_k^\theta(x, \cdot)\right)}_{\text{rate-freezing error}} + \underbrace{D_{\mathrm{TV}}\left(\bar{P}_k^\theta(x, \cdot), \widehat{P}_k^\theta(x, \cdot)\right)}_{\text{multiple-jump error}}.$$

We bound these two terms separately below.

Multiple-jump error. The implemented sampler allows at most one jump per coordinate during $[t_k, t_{k+1}]$, while the frozen-rate process may make multiple jumps. Let N_i denote the number of actual jumps in coordinate i, and let $\widetilde{N}_i \sim \mathrm{Poisson}(r_i(x, t_k)\Delta_k)$ denote the number of candidate events under the constant frozen rate. Since death events are suppressed at zero, N_i is stochastically dominated by $\widetilde{N}_i$.

To bound $D_{\mathrm{TV}}(\bar{P}_k^\theta(x,\cdot), \widehat{P}_k^\theta(x,\cdot))$, the second term above, note that the frozen process and the sampler agree whenever every coordinate makes at most one jump. Therefore,

$$D_{\mathrm{TV}}\left(\bar{P}_k^\theta(x,\cdot), \widehat{P}_k^\theta(x,\cdot)\right) \le \mathbb{P}\left(\bigcup_{i=1}^d \{N_i \ge 2\}\right) \le \sum_{i=1}^d \mathbb{P}(\widetilde{N}_i \ge 2)$$
$$\le \frac{1}{2}\sum_{i=1}^d \mathbb{E}\Big[\widetilde{N}_i(\widetilde{N}_i - 1)\Big] = \frac{\Delta_k^2}{2}\sum_{i=1}^d r_i(x,t_k)^2,$$

where $\mathbb{P}(N \ge 2) \le \mathbb{E}[N(N-1)]/2 = m^2/2$ for $N \sim \mathrm{Poisson}(m)$.

Thus the local multiple-jump error is $O(\Delta_k^2)$ and scales with the squared jump-rate sum, $\sum_{i=1}^d r_i(x,t_k)^2$. The bound therefore suggests using smaller time steps Δ_k in regions where the squared jump-rate sum is large.

Rate-freezing error. To bound $D_{\mathrm{TV}}(P_k^\theta(x,\cdot), \bar{P}_k^\theta(x,\cdot))$, the first term above, let Y_u denote the exact learned process after elapsed time $u \in [0,\Delta_k]$ from t_k, with $Y_0 = x$. Comparing the exact and frozen-rate processes over the same interval, their TV difference is bounded by the accumulated difference in their jump rates. Applying the Lipschitz bound,

$$D_{\mathrm{TV}}\left(P_k^\theta(x,\cdot), \bar{P}_k^\theta(x,\cdot)\right) \le \int_0^{\Delta_k} \mathbb{E}\left[\|\nu_\theta(Y_u, t_k+u) - \nu_\theta(x,t_k)\|_1\right] du$$
$$\le \int_0^{\Delta_k} \left[L_t u + L_x\, \mathbb{E}\|Y_u - x\|_1\right] du$$
$$\le \frac{\Delta_k^2}{2}(L_t + AL_x).$$

The last inequality uses $a_t^\theta \le A$, where $a_t^\theta(x,t) = \sum_{i=1}^d r_i(x,t)$. Since the CTMC only makes local transitions $x \to x \pm e_i$, its displacement from x is bounded by the number of jumps, giving $\mathbb{E}\|Y_u - x\|_1 \le Au$. Thus the local rate-freezing error is $O(\Delta_k^2)$ and grows with $L_t + AL_x$.

Combining the two contributions,

$$D_{\mathrm{TV}}\left(P_k^\theta(x,\cdot), \widehat{P}_k^\theta(x,\cdot)\right) \le \frac{\Delta_k^2}{2}\left[\sum_{i=1}^d r_i(x,t_k)^2 + L_t + AL_x\right].$$

Thus the one-step discretization error is $O(\Delta_k^2)$. Using TV contraction under Markov kernels and summing over the K intervals,

$$D_{\mathrm{TV}}(p_\tau^\theta, \widehat{p}_\tau^\theta) \le \frac{1}{2}\sum_{k=0}^{K-1} \Delta_k^2 \left[\mathbb{E}_{X_{t_k}\sim p_{t_k}^\theta} \sum_{i=1}^d r_i(X_{t_k}, t_k)^2 + L_t + AL_x\right].$$

For the uniform grid used in Section 2.2, $\Delta_k = \Delta = \tau/K$. Using the global rate bound, $\sum_i r_i^2 \le (\sum_i r_i)^2 \le A^2$,

$$D_{\mathrm{TV}}(p_\tau^\theta, \widehat{p}_\tau^\theta) \le \frac{\tau^2}{2K}\left(A^2 + L_t + AL_x\right) = O(K^{-1}).$$

Thus the sampler is globally first order. The local error increases with the squared jump-rate sum $\sum_{i=1}^d r_i(x,t_k)^2$ and the rate-variation term $L_t + AL_x$, suggesting the use of smaller time steps Δ_k when either term is large.

A.4.4 Overall error bound

Combining the endpoint truncation, rate estimation, and uniform-grid discretization bounds gives

$$D_{\mathrm{TV}}(p_1, \widehat{p}_\tau^\theta) \le \underbrace{\varepsilon_t \, \mathbb{E}_\pi \|X_1 - X_0\|_1}_{\text{endpoint truncation}} + \underbrace{\sqrt{\frac{\tau}{2} \left[\mathcal{L}_{\mathrm{train}}(\theta) - \mathcal{L}^*_{\mathrm{train}}\right]}}_{\text{rate estimation}} + \underbrace{\frac{\tau^2}{2K} \left(A^2 + L_t + A L_x\right)}_{\text{discretization}}.$$

Thus the overall terminal error separates into an $O(\varepsilon_t)$ endpoint term, a term controlled by the excess rate-matching loss, and an $O(K^{-1})$ discretization term. If the endpoint moments and the rate bounds A, L_x, and L_t remain bounded, then the bound vanishes as $\varepsilon_t \to 0$, $\mathcal{L}_{\mathrm{train}}(\theta) - \mathcal{L}^*_{\mathrm{train}} \to 0$, and $K \to \infty$.

B Simulation

Unless otherwise stated, all experiments were run on a single NVIDIA RTX 4090 GPU with 24GB memory.

Figure 3 visualizes representative intermediate samples along each model's native sampling trajectory in the simulation. Since different methods use different native time parameterizations and different sampling mechanisms, this figure is qualitative and is intended only to show how generated samples evolve under each fitted model's own dynamics.

To compare intermediate bridge behavior across models, we construct, for each method and each coordinate, an empirical family of one-dimensional intermediate marginals indexed by a common progress variable $s \in [0, 1]$. For a fixed value of s, we repeatedly sample endpoint pairs $(x_0^{(m)}, x_1^{(m)})$, generate an intermediate state $X_s^{(m)}$ using the corresponding model-specific bridge or forward noising distribution, and estimate the marginal pmf by

$$\widehat{p}_s^{(i)}(z) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}\left\{X_s^{(i,m)} = z\right\}, \qquad z \in \{0, 1, \ldots, C_{\max}\}.$$

The heatmaps in Figure 4 plot $\widehat{p}_s^{(i)}(z)$ as a function of count value z and common progress s.

For count-FM, the conditional bridge is

$$X_t^{(i)} = x_0^{(i)} + \mathrm{sgn}\left(x_1^{(i)} - x_0^{(i)}\right) B_t^{(i)}, \qquad B_t^{(i)} \sim \mathrm{Binomial}\left(\left|x_1^{(i)} - x_0^{(i)}\right|, t\right),$$

and we set $s = t$. For count-FM-OT, we use the same conditional bridge, but the endpoints (x_0, x_1) are first paired using the minibatch OT coupling from Section 2.3. We do not include Poisson-JUMP in Figure 4, because its Poisson-source jump construction does not provide a directly comparable one-parameter bridge or forward noising family under the same common-progress normalization.

For the remaining baselines, since each method uses a different forward process and parameterization, we convert each native time variable to a common progress variable $s \in [0, 1]$ by monotone rescaling of a model-specific progress statistic $\rho(t)$,

$$s = \frac{\rho(t) - \rho(0)}{\rho(1) - \rho(0)}.$$

Here $\rho(t)$ measures source-to-target progress along the native path of each method. This places all comparison methods on the same source-to-target scale. Under this normalization, the count-FM bridges evolve gradually across the full range of s, whereas the categorical-state diffusion- and flow-based baselines typically become target-like much earlier, reflecting a sharper transition in count space.

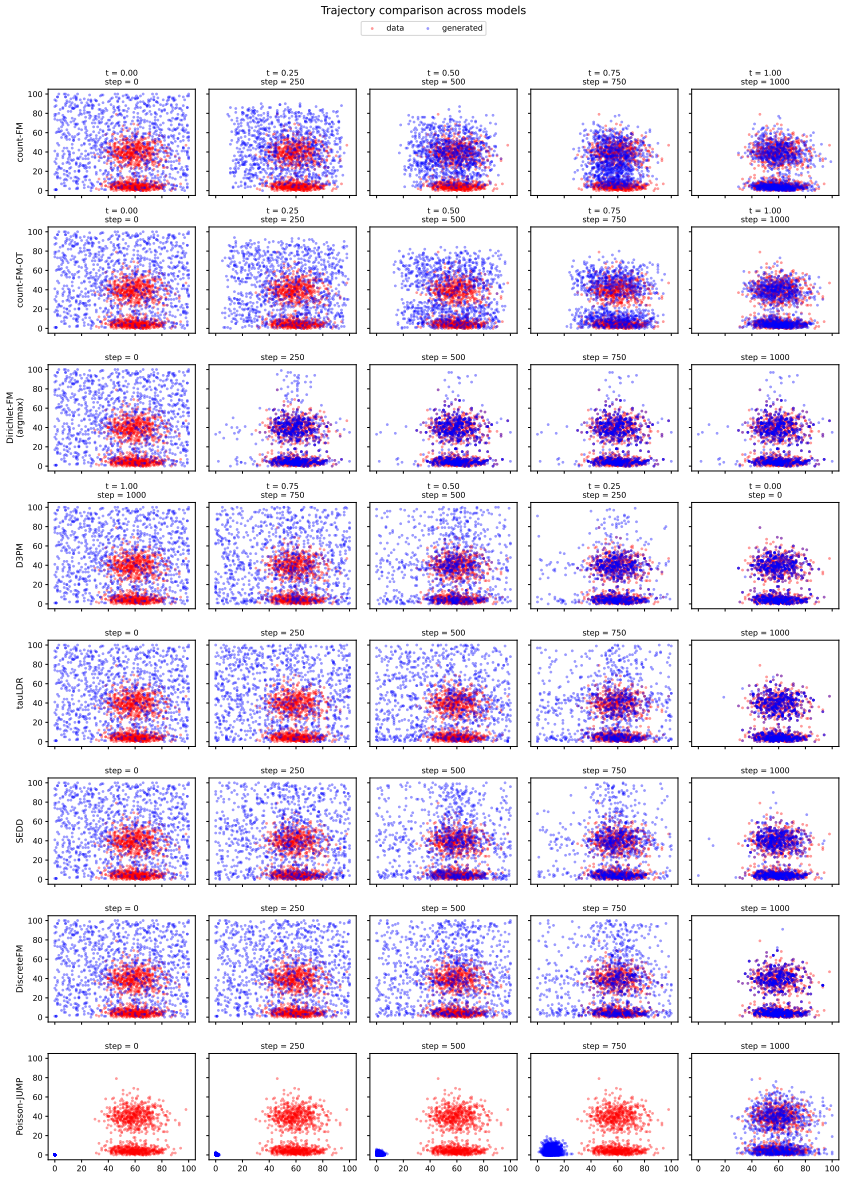

Figure 3: **Representative intermediate samples from the native sampling trajectories of different models in the simulation.** Red points show target samples and blue points show generated samples. Columns correspond to increasing native sampling time or sampling step within each method's own generation procedure.

C OT coupling and sampling efficiency

Figure 5 compares the sampling efficiency of count-FM and count-FM-OT in both the simulation and scRNA experiments. We plot W_2 and $\mathrm{MMD}^2_{\mathrm{RBF}}$ against the number of function evaluations (NFE) and wall-clock runtime. In the simulation, count-FM-OT shows a clear efficiency advantage across most computational budgets. On the scRNA task, the advantage is smaller but still visible at lower budgets, while the two methods become very similar as the budget increases.

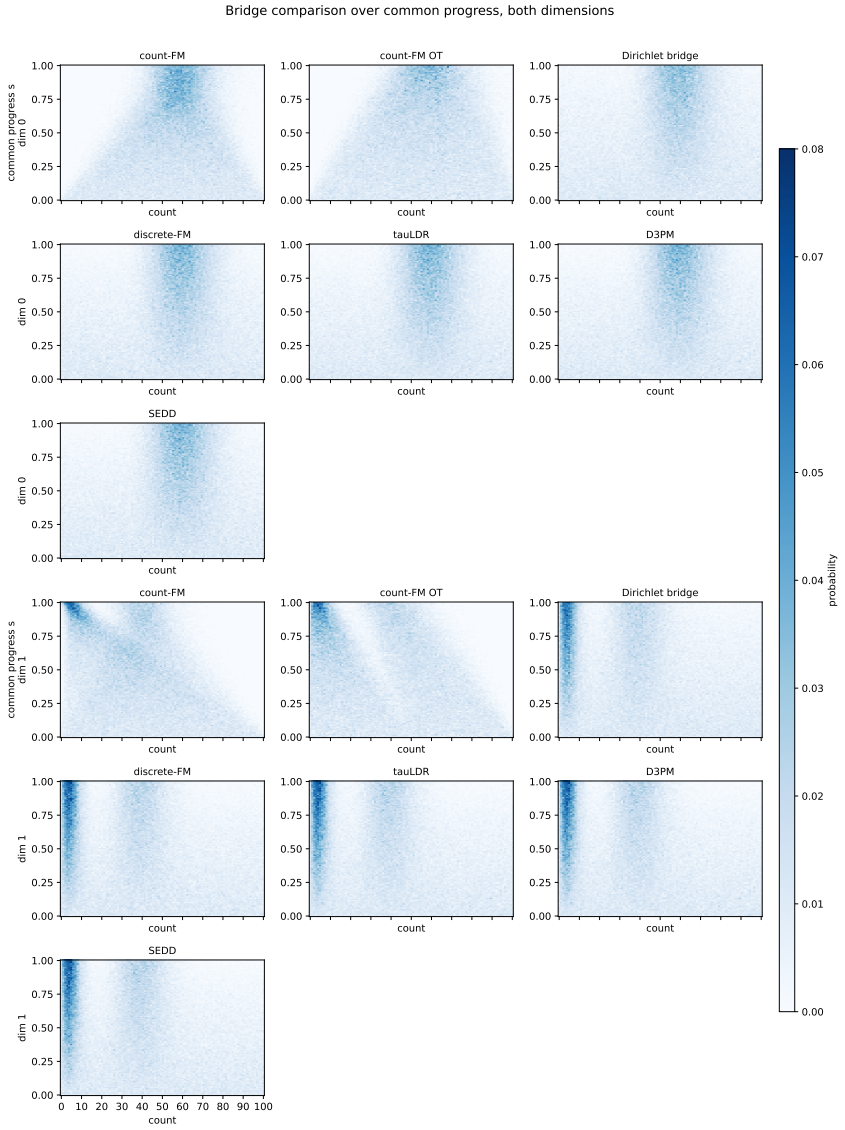

Figure 4: **Marginal bridge distributions for both coordinates.** They are shown under a common progress variable s. Intermediate marginals are estimated by Monte Carlo sampling from each model's bridge or forward noising distribution after mapping to the common progress scale. Count-FM evolves gradually across s, while categorical-state baselines become target-like early. count-FM-OT uses the same bridge as count-FM but with OT-coupled endpoints, yielding a visibly straighter transition path.

D Applications

D.1 single-cell RNA-seq generation and transport

For unconditional generation, we perform paired comparisons across five matched generation seeds. Differences are baseline minus count-FM, so positive values favor count-FM. We report the paired standardized effect size $d_z = \overline{\Delta}/s_\Delta$ and Holm-corrected p-values [Holm, 1979].

Although some gains are modest, all differences remain significant after Holm correction.

PCA-space W_2 and $\mathrm{MMD}^2_{\mathrm{RBF}}$ measure global distributional agreement rather than biological fidelity directly. We therefore select marker genes using the training data. For each annotated cell type, we

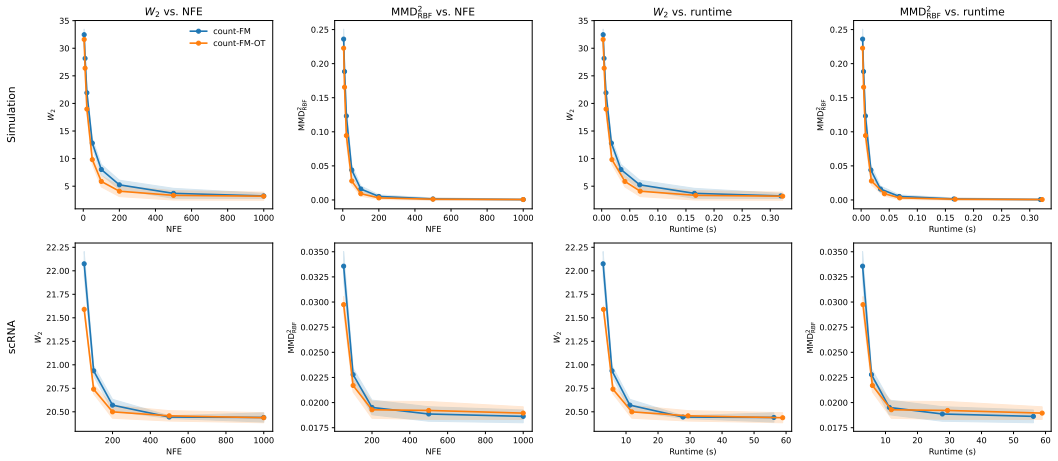

Figure 5: **Sampling efficiency comparison between count-FM and count-FM-OT in the simulation and scRNA experiments.** We plot W_2 and $\mathrm{MMD}^2_{\mathrm{RBF}}$ against the number of function evaluations (NFE) and wall-clock runtime. In the simulation, count-FM-OT reaches a given quality level with smaller computational budget. On the scRNA task, the same trend is present but weaker, and the difference becomes small at larger budgets.

Table 4: **Paired comparison with count-FM on the Dentate Gyrus dataset.**

Baseline	ΔW_2	d_z	Holm p	$\Delta\mathrm{MMD}^2_{\mathrm{RBF}}$	d_z	Holm p
scDiffusion	0.0886	1.27	0.0466	0.00401	6.23	3.39×10^{-4}
scLDM	0.2823	7.20	2.61×10^{-4}	0.00175	2.47	0.00525
scVI	0.2303	3.91	0.00188	0.00439	6.74	3.39×10^{-4}
DCM	3.1095	50.26	1.88×10^{-7}	0.01111	21.50	5.60×10^{-6}
CFGen	1.7700	21.98	4.10×10^{-6}	0.01766	16.89	1.17×10^{-5}

retain the 10 genes with the largest log-normalized mean-expression difference from all other cell types, yielding 99 unique markers.

Table 5: **Marker-gene recovery on the Dentate Gyrus dataset.**

Model	Marker correlation ↑	Marker MAE ↓
count-FM	**0.9959**	**0.0652**
scLDM	0.9941	0.0697
scDiffusion	0.9927	0.0710
scVI	0.9934	0.0778
DCM	0.9710	0.1296
CFGen	0.9926	0.1534

All methods achieve high marker-expression correlations, while count-FM obtains the lowest marker MAE, indicating the most accurate recovery of cell-type-relevant expression levels. Appendix Figures 6 and 7 complement the main-text trajectory and fate summaries by visualizing the temporal evolution of the transport process on both the training and test sets.

D.2 Additional results for hippocampal hc-3

This appendix provides expanded qualitative diagnostics for the hc-3 conditional-generation experiment. For generative models, all plotted summaries are estimated from 100 generated samples per held-out covariate. Appendix Figure 8 shows bin-wise mean responses across signed position for all models, with neurons grouped by brain region. The MLP mean regressor, Poisson MLP, and count-FM with $w = 1$ all recover the broad spatial response pattern. Count-FM with $w = 0$ lacks

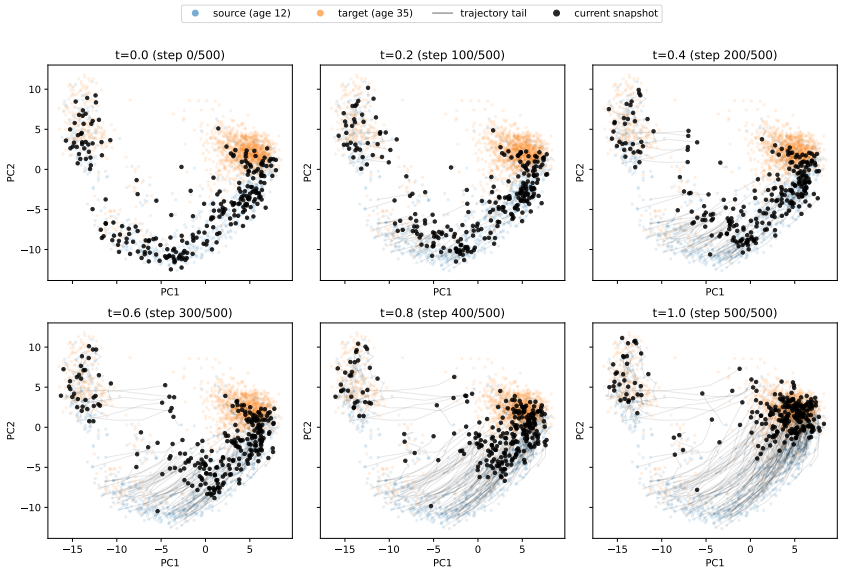

Figure 6: **Transition snapshots for training.** As time increases, generated cells move progressively from the P12 source manifold toward the P35 target manifold in PCA space. Light colored points show the source and target references, gray curves show trajectory tails, and black points show the current generated snapshot.

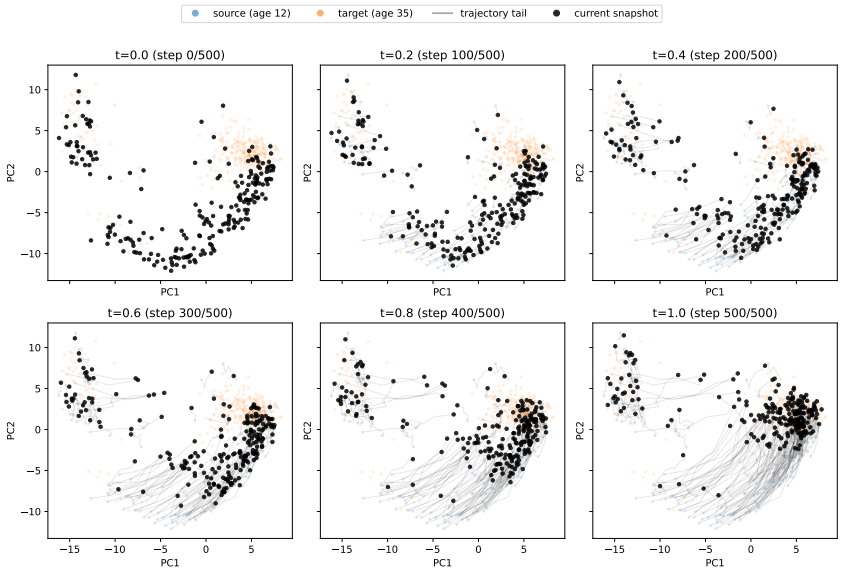

Figure 7: **Transition snapshots for testing.** The same progressive transport pattern is observed on held-out cells, with trajectories moving smoothly from the P12 manifold toward the P35 manifold.

conditional information and only captures marginal firing-rate differences across neurons, while $w = 2$ sharpens location-specific responses but is less well calibrated to the true mean pattern.

Appendix Figure 9 shows the corresponding population correlation structure using the same neuron set. Since the MLP mean regressor is deterministic and is not a generative count model, it is omitted from this dependence comparison. Count-FM with $w = 1$ most closely matches the true dependence structure, while the Poisson MLP underestimates population correlations. The unconditional model ($w = 0$) still captures some marginal dependence structure, whereas $w = 2$ amplifies the correlation pattern but is less well calibrated. These are consistent with Table 3, where $w = 1$ gives the best

overall generative fidelity, whereas $w = 2$ mainly sharpens the response pattern at the cost of calibration.

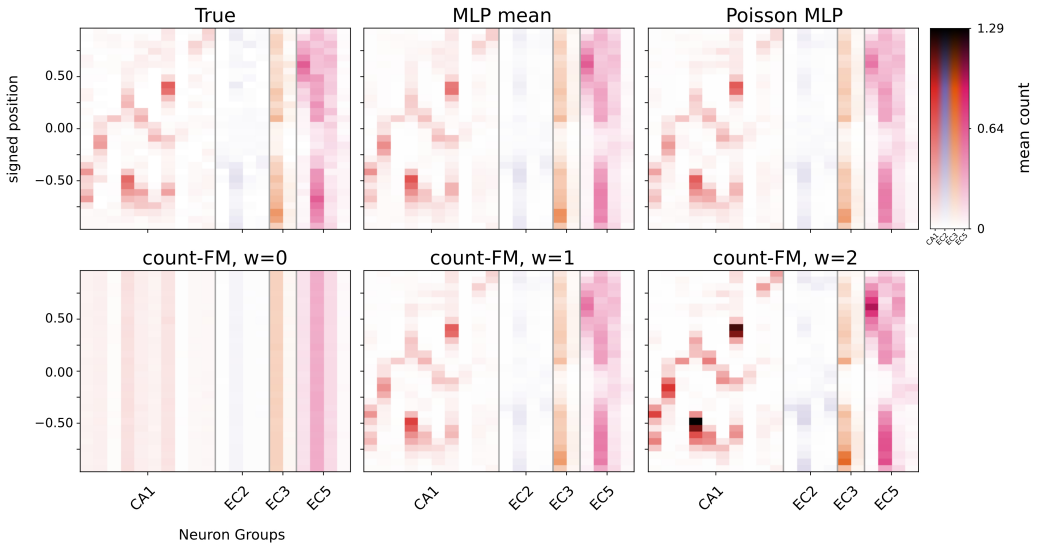

Figure 8: **Mean-response comparison for the hc-3 conditional-generation task.** Each panel shows bin-wise mean counts across signed position, with neurons grouped by region. For generative models, summaries are estimated from 100 generated samples per held-out covariate. The MLP mean regressor, Poisson MLP, and count-FM with $w = 1$ recover the broad spatial response pattern. Count-FM with $w = 0$ lacks spatial specificity, while $w = 2$ sharpens location-specific responses but reduces calibration.

D.3 Conditional generation of piriform cortex spike trains

We also evaluate count-FM on conditional generation of multichannel piriform cortex (PCx) spike trains using the public odor-response dataset of Bolding and Franks [2017], Bolding et al. [2020]. The dataset contains processed, spike-sorted extracellular recordings together with simultaneously recorded respiration traces from head-fixed mice. We focus on the PCx recording from session 141208-2 (bank 2), which contains 480 trials. Each trial is aligned to inhalation onset, spikes are binned into 10 ms intervals over the window $[-0.5, 1.5]$ s, and each sample is represented as a time-by-neuron count matrix with 200 time bins and 84 neurons [Miura et al., 2012]. The conditioning variables include time, a respiration-waveform covariate, odor presence, odor identity, and binarized inhalation state. We use a trial-level 80/20 train-test split stratified by odor identity, giving 384 training trials and 96 held-out test trials.

Figure 10 shows mean responses over held-out odor-3 trials. The mean MLP captures the coarse response shape but is smoother than the held-out mean, while the Poisson MLP is a stronger count baseline but remains comparatively diffuse. Count-FM recovers the odor- and respiration-dependent structure better. Increasing guidance from $w = 1$ to $w = 2$ sharpens the aligned response pattern, but also increases its amplitude. After aligning the spiking activity with covariates (upper panel in Figure 10) by time, we can see that the neural activity is strongly correlated with the inhalation state (lower firing rate during inhalation), especially when the odor is present. Figure 11 shows the same guidance tradeoff under a fixed held-out covariate. Here, the mean MLP is not shown because it is not generative.

To quantify what we observed, we calculate different metrics and Table 6 summarizes held-out metrics over 5 replicated runs, each with a new odor-stratified 80/20 train-test split and retraining under different random seeds. Among the count-FM settings, $w = 1$ gives the best overall tradeoff, with the lowest mean RMSE and 95th-percentile error, while remaining close to the Poisson MLP baseline in matched cosine and mean-count ratio. By contrast, $w = 2$ gives the highest matched cosine among count-FM settings, but substantially worse mean RMSE and a larger mean-count ratio, indicating stronger condition alignment at the cost of amplitude distortion.

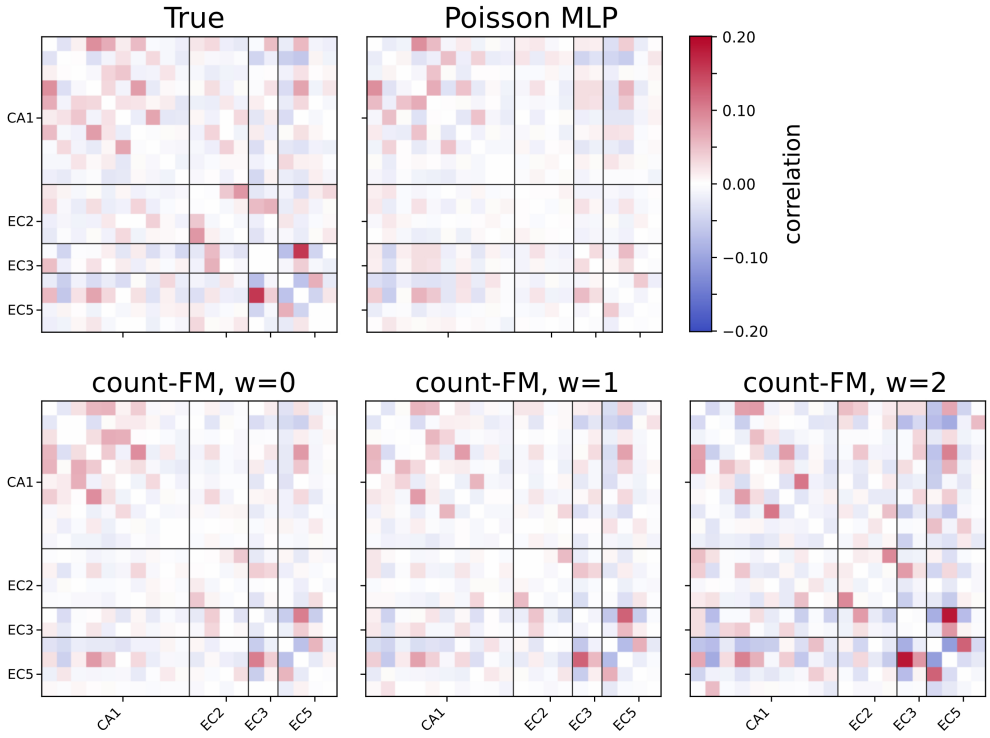

Figure 9: **Population correlation comparison for the hc-3 conditional-generation task.** Panels show correlation matrices for the active-neuron set. These are estimated from 100 generated samples per held-out covariate. Count-FM with $w = 1$ most closely matches the true dependence structure, while the Poisson MLP underestimates population correlations. Count-FM with $w = 0$ still captures some marginal dependence structure, and stronger guidance ($w = 2$) amplifies the correlation pattern but is less well calibrated.

Table 6: **Held-out conditional generation quality on the PCx task.** Results are mean ± standard deviation over 5 random seeds, each with a new odor-stratified train-test split and model retraining. count-FM with $w = 1$ gives the best overall tradeoff among count-FM settings, while $w = 2$ improves matched cosine but over-amplifies spike counts.

Model	Mean RMSE ↓	Matched cosine ↑	Mean-count ratio	95th-percentile error ↓
MLP mean	1.264 ± 0.248	**0.800 ± 0.041**	0.704 ± 0.100	0.487 ± 0.017
Poisson MLP	1.090 ± 0.071	0.762 ± 0.018	1.073 ± 0.058	0.311 ± 0.069
count-FM, $w = 0$	2.090 ± 0.255	0.407 ± 0.074	1.300 ± 0.429	0.322 ± 0.093
count-FM, $w = 1$	**1.082 ± 0.056**	0.760 ± 0.019	**1.037 ± 0.078**	**0.297 ± 0.071**
count-FM, $w = 2$	1.744 ± 0.342	0.792 ± 0.022	1.427 ± 0.188	0.336 ± 0.029

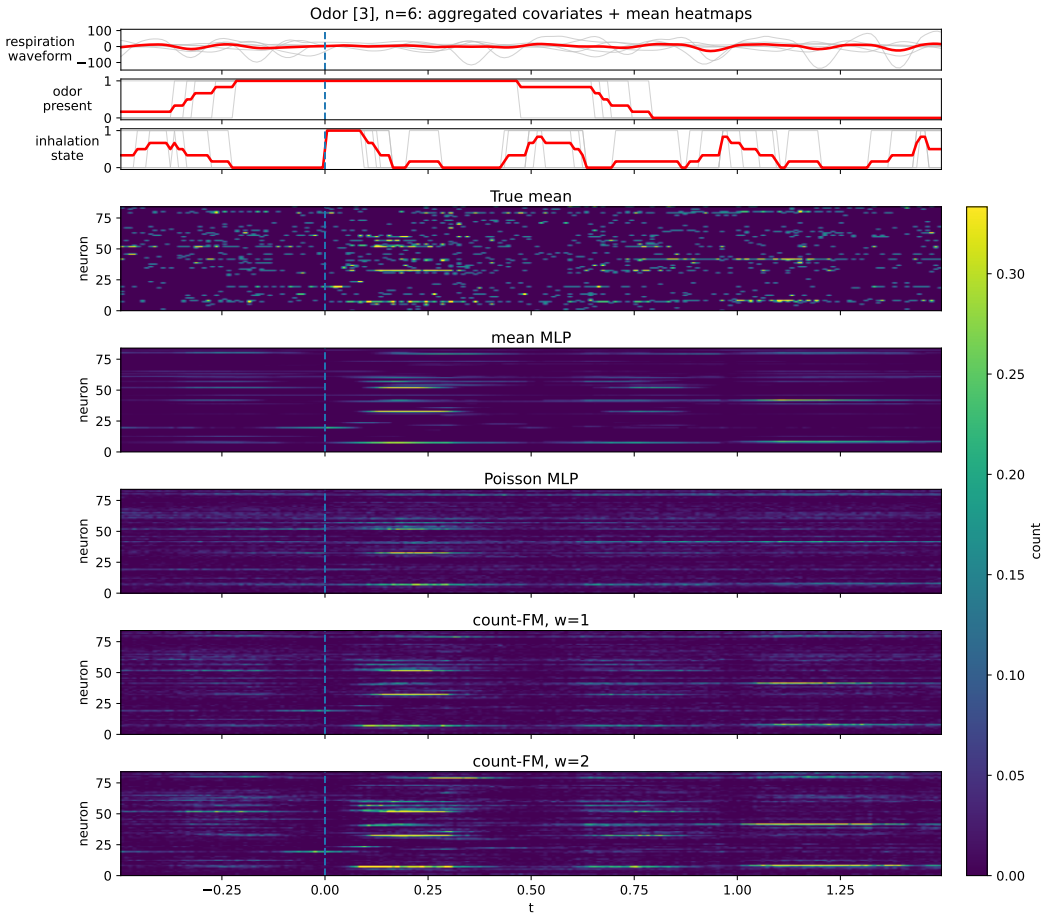

Figure 10: **Conditional generation on held-out odor-3 trials.** Upper panel shows conditioning covariates over time, with gray traces for individual held-out trials and red traces for trial averages. Lower panel shows neuron-by-time mean responses for the held-out data, the MLP mean regressor, Poisson MLP, and count-FM with $w = 1$ and $w = 2$. The MLP mean regressor is overly smooth and Poisson MLP remains diffuse, while count-FM better recovers localized odor- and respiration-dependent responses. Increasing guidance to $w = 2$ sharpens the response pattern but over-amplifies spike counts, reducing amplitude calibration.

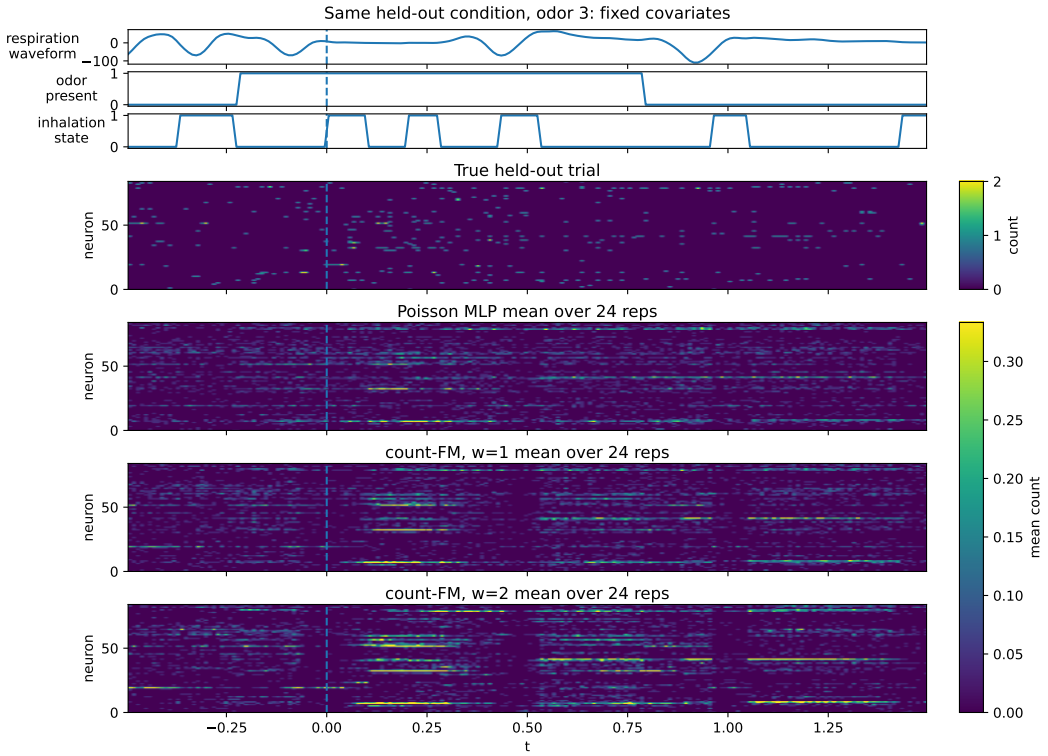

Figure 11: **Generated responses under the same held-out covariate trajectory.** We compare the true held-out trial with the mean response from the Poisson MLP regressor over 24 repetitions, and with the count-FM mean responses over 24 repetitions at $w = 1$ and $w = 2$. Compared with the Poisson MLP regressor, count-FM produces a more structured condition-aligned response. Increasing guidance from $w = 1$ to $w = 2$ sharpens the shape further, but also inflates the response amplitude.

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: The abstract and introduction summarize the proposed method, experiments, and scope of the claims.

   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: Limitations and future directions are discussed in Section 5.

   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: Derivations of the bridge rates and KL objectives are provided in Appendix A.

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: Datasets, metrics and model settings are described in the main text and appendix.

   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: The experiments use public datasets and code is attached as Supplementary Material.

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: All reported.

   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: See tables in the main text.

   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: provided in the appendix.

   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: public datasets and does not involve new human-subject data collection.

   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: [No]

    Justification: This is a methodological paper.

    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: no high-risk models or datasets.

    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: Public dataset.

    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: [N/A]

    Justification: The paper does not release a new dataset or benchmark asset.

    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: no crowdsourcing or new 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: no 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: LLMs are not part of the core method or experiments.

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.