COUNT BRIDGES ENABLE MODELING AND DECONVOLVING TRANSCRIPTOMIC DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Many modern biological assays, including RNA sequencing, yield integer-valued counts that reflect the number of RNA molecules detected. These measurements are often not at the desired resolution: while the unit of interest is typically a single cell, many RNA sequencing and imaging technologies produce counts aggregated over sets of cells. Although recent generative frameworks such as diffusion and flow matching have been extended to non-Euclidean and discrete settings, it remains unclear how best to model integer-valued data or how to systematically deconvolve aggregated observations. We introduce Count Bridges, a stochastic bridge process on the integers that provides an exact, tractable analogue of diffusion-style models for count data, with closed-form conditionals for efficient training and sampling. We extend this framework to enable direct training from aggregated measurements via an Expectation-Maximization-style approach that treats unit-level counts as latent variables. We demonstrate state-of-the-art performance on integer distribution matching benchmarks, comparing against flow matching and discrete flow matching baselines across various metrics. We then apply Count Bridges to two large-scale problems in biology: modeling single-cell gene expression data at the nucleotide resolution, with applications to deconvolving bulk RNA-seq, and resolving multicellular spatial transcriptomic spots into single-cell count profiles. Our methods offer a principled foundation for generative modeling and deconvolution of biological count data across scales and modalities.

1 INTRODUCTION

Integer-valued counts are a fundamental product of scientific measurements because of the discrete nature of molecules. Modern biological assays yield massive streams of count data: RNA-seq read counts, fluorescence imaging molecule counts, and mass cytometry ion counts (Klein et al., 2015; Raj et al., 2008; Bendall et al., 2011). However, these measurements are often aggregated over multiple individual units, obscuring the fine-grained patterns underlying these natural phenomena. Transcriptomics technologies exemplify this challenge, with technologies such as Visium capturing 10-50 cells per spot (Ståhl et al., 2016) and bulk RNA-seq aggregating thousands to millions of cells per readout, yielding averages rather than high-resolution details. Deconvolving these aggregates into single-cell profiles is critical for the precise mapping of cellular heterogeneity, cell-cell interactions, and tissue architecture (Moses & Pachter, 2022; Armingol et al., 2021). The challenge is twofold: building generative models that respect the integer nature of counts and extending these models to infer unit-level profiles from aggregated observations.

Recent developments in generative modelling only partially adresss the problem. Discrete diffusion models (Austin et al., 2021; Lou et al., 2023) treat counts as unordered categories through masking or uniform noise. Blackout Diffusion (Santos et al., 2023), the only count-specific approach, uses pure-death processes that cannot transport between arbitrary distributions. The biological deconvolution literature on the other hand focuses on deconvolving cell-type (cluster-level) proportions (Kleshchevnikov et al., 2022; Cable et al., 2022; Li et al., 2023b), rather than unit-level count profiles. Thus, there is need for a framework that respects the integer and ordinal structure of counts, enables transport between arbitrary distributions, and can systematically deconvolve aggregated observations.

We introduce Count Bridges: a stochastic bridge process on $\mathbb{Z}^d$ using Poisson birth-death dynamics. This yields closed-form conditionals for exact sampling and extends naturally to deconvolution via an

EM algorithm treating unit-level counts as latent. The birth-death mechanism allows transport between arbitrary integer-valued distributions while preserving the ordinal structure, as both increments and decrements respect the natural ordering of counts. We show that Count Bridges outperform existing methods on synthetic benchmark datasets and scale more favorably to high-dimensional settings. We then showcase Count Bridges on two real-world biological applications centered on deconvolution: nucleotide-resolution single-cell RNA-sequence modeling for bulk RNA-seq deconvolution and reference-free spatial transcriptomic deconvolution. The anonymized codebase is available here.

2 Background on diffusion models

We present diffusion models as a time–indexed family of *bridge kernels* connecting $X_0 \sim p_{\mathrm{data}}$ and $X_1 \sim \nu$, where ν is a source distribution (often Gaussian). We will use two kernels: (i) the *global bridge* $K_{t|1}(x_1, x_0) \equiv \mathsf{Law}(X_t \mid X_1{=}x_1, X_0{=}x_0)$, used to draw training tuples; and (ii) the *local bridge* $K_{s|t}(x_t,\, x_0) \equiv \mathsf{Law}(X_s \mid X_t{=}x_t, X_0{=}x_0)$ for $0 \le s \le t \le 1$, used to compose steps along a time grid. We train a denoiser q_θ that allows approximately sampling the posterior,

$$\tilde{X}_0 \sim q_\theta(x_t, t) \approx \mathsf{Law}(X_0 \mid X_t{=}x_t),$$

using pairs (t, X_t) from the global bridge. For sampling, pick $1 = t_K > \cdots > t_0 = 0$, draw $X_1 \sim \nu$, set $X_{t_K} \leftarrow X_1$, and for $k = K{-}1, \ldots, 0$ sample

$$\tilde{X}_0^{(k+1)} \sim q_\theta(X_{t_{k+1}}, t_{k+1}), \qquad X_{t_k} \sim K_{t_k|t_{k+1}}(X_{t_{k+1}},\, \tilde{X}_0^{(k+1)}). \tag{1}$$

So the sampling process cannot drift out of the training distribution and $p_s(x_s)$ is equivalently:

$$\int K_{s|1}(x_s|x_1, x_0)\nu(x)dx_1 = \int K_{s|t}(x_s \mid x_t, x_0)\, p_{0|t}(x_0 \mid x_t)\, K_{t|1}(x_t \mid x_1, x_0)\, \nu(x_1)\, dx_0\, dx_t\, dx_1.$$

Different problem classes require different bridge families $K_{\cdot|\cdot}$ and different methods to approximate the posterior, but the general training/sampling scheme above is unchanged.

2.1 Diffusion as a bridge between noise and data

First, let us consider a process $(X_t)_{t\in[0,1]}$ of the following form

$$X_t = \alpha_t X_0 + B_{\sigma_t}, \tag{2}$$

where $(B_t)_{t\in[0,1]}$ is a d-dimensional Brownian motion, α_t is a non-increasing process and σ_t is a non-decreasing process. In addition, we assume that $X_0 \sim p_{\mathrm{data}}$. Note that $\alpha_0 = 1$ and $\sigma_0 = 0$.

First, we want to define a process that interpolates smoothly between $X_0 \sim p_{\mathrm{data}}$ and X_1 given by another distribution as in Peluchetti (2023); Albergo et al. (2023); Delbracio & Milanfar (2023); Liu et al. (2022; 2023). We have the following proposition defining the global and local bridge.

Proposition 2.1. *Let $(X_t)_{t\in[0,1]}$ be given by equation 2. Now, consider $(X_s)_{s\in[0,t]}$ conditioned by $X_t = x_t$ and $X_0 = x_0$. Then, we have that*

$$X_s \overset{d}{=} \alpha_s(1 - r_{s,t})X_0 + \frac{\alpha_s}{\alpha_t} r_{s,t} X_t + \sigma_s(1 - r_{s,t})^{1/2} Z, \tag{3}$$

where $Z \sim \mathcal{N}(0, \mathrm{Id})$ is independent of X_0, X_t and $r_{s,t} = \frac{\alpha_t^2 \sigma_s^2}{\alpha_s^2 \sigma_t^2}$.

We can use this as the global bridge for training by setting $t = 1$, and as a local bridge for sampling using t and s iteratively. Note that if $X_1 \sim \mathcal{N}(0, \mathrm{Id})$, $\alpha_1 = 0$ and $\sigma_1 = 1$ then we get that

$$X_t \overset{d}{=} \alpha_t X_0 + \sigma_t Z. \tag{4}$$

Furthermore, our equation 3 recovers the interpolation described in Albergo et al. (2023) with the identification $\alpha_t \to \alpha_t(1 - r_t)$, $\frac{\alpha_t}{\alpha_1} r_t \to \beta_t$ and $\sigma_t(1 - r_t)^{1/2} \to \gamma_t$.

2.2 Sampling the Posterior

In this paradigm the bridge is only the first of two choices that define the model. We also have to choose how to model the posterior $X_0|X_t, t$. There are two core options: we can use differential equations to model the posterior in the limit of small steps or we can focus more directly on modeling the posterior. In Euclidean space, the former lets us learn a simple conditional expectation, whereas the latter always requires a distribution model.

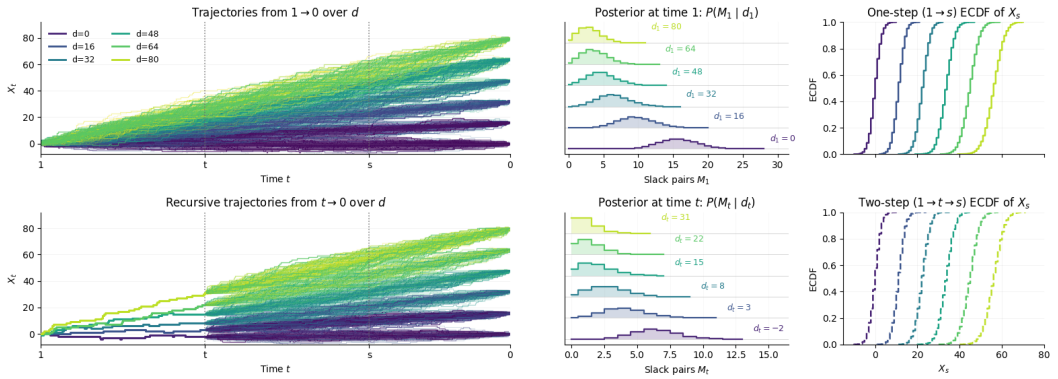


Figure 1: *Left:* Sample paths for several endpoint gaps d_1 (top). Fixing the prefix $[0, t]$ resample $(t, 1]$ by the recursive kernel (bottom). *Middle:* Bessel slack posteriors at initial and intermediate times. The slack M_t concentrates near 0 as $|d|$ grows. *Right:* ECDFs of X_s from a one–step kernel $(1 \to s)$ and a two–step kernel $(1 \to t \to s)$ are indistinguishable, confirming composition.

Infinitesimal. For $X_t = \alpha_t X_0 + B_{\sigma_t}$ and a small step $\delta > 0$, the local bridge yields

$$X_t \approx X_{t+\delta} + \delta\, b(X_{t+\delta}, t+\delta) + \sqrt{\delta}\, \epsilon_{t,\delta},$$

where the *drift* b and *noise* A are defined by the conditional moments

$$\mathbb{E}[X_t \,|\, X_{t+\delta}{=}x] = x + \delta\, b(x, t+\delta) + o(\delta), \quad \mathrm{Cov}[X_t \,|\, X_{t+\delta}{=}x] = \delta\, A(x, t+\delta) + o(\delta),$$

and $\epsilon_{t,\delta}$ satisfies $\mathbb{E}[\epsilon_{t,\delta} \,|\, X_{t+\delta}{=}x] = 0$, $\mathrm{Cov}[\epsilon_{t,\delta} \,|\, X_{t+\delta}{=}x] = A(x, t+\delta)$. By linearity in X_0,

$$b(x,t) = B_1(t)\, x + B_2(t)\, \mu_{0|t}(x) + b_0(t), \qquad A(x,t) = A_0(t),$$

depends on the posterior $\mathsf{Law}(X_0 \,|\, X_t{=}x)$ only via its mean $\mu_{0|t}(x) = \mathbb{E}[X_0 \,|\, X_t{=}x]$. Hence a point denoiser $\hat{x}_\theta(x,t) \approx \mu_{0|t}(x)$ (equivalently a score/velocity) is sufficient here (Song et al., 2020).

Distributional. Following De Bortoli et al. (2025) we can learn the *conditional law* $q_\theta(\,\cdot \mid x_t, t) \approx \mathsf{Law}(X_0 \mid X_t{=}x_t)$, which we directly plug into equation 1. The distributional perspective is particularly powerful when the infinitesimal perspective fails to admit a simplification to the conditional expectation, which motivates our use of the distributional approach for count bridges (see Sec. 3.2). Note that in categorical discrete settings, all approaches are distributional since they are based on cross-entropy losses, see Campbell et al. (2022); Austin et al. (2021); Shi et al. (2024); Sahoo et al. (2024) for instance.

3 Count Bridges

3.1 An integer bridge between distributions

Mirroring Sec. 2, we seek to develop a bridge for integer data. We let $\lambda_\pm : [0,1] \to \mathbb{R}_{\geq 0}$ be rate functions with cumulants $\Lambda_\pm(t) = \int_0^t \lambda_\pm(s)\, \mathrm{d}s$ and let $w(t) = \frac{\Lambda_+(t)+\Lambda_-(t)}{\Lambda_+(1)+\Lambda_-(1)}$. We draw independent Poisson processes $(B_t)_{t\in[0,1]} \sim \mathrm{Poi}(\Lambda_+)$ and $(D_t)_{t\in[0,1]} \sim \mathrm{Poi}(\Lambda_-)$, then

$$X_t = X_0 + B_t - D_t. \tag{5}$$

We denote $d_t = X_t - X_0$, the total number of jumps $N_t = B_t + D_t$, and the *slack* variable $M_t = \min(B_t, D_t)$. Any two of these variables identify the others:

$$N_t = |d_t| + 2M_t, \qquad B_t = \tfrac{1}{2}(N_t + d_t), \qquad D_t = N_t - B_t. \tag{6}$$

With these results in hand we can derive the counterpart of Proposition 2.1 in this setting.

Proposition 3.1. *Let $(X_t)_{t\in[0,1]}$ be given by equation 5. Now, consider $(X_s)_{t\in[0,t]}$ conditioned by $X_t = x_t$ and $X_0 = x_0$. Then, we have that*

$$X_s \overset{d}{=} X_0 + B_s - D_s, \tag{7}$$

where

$$N_s \mid N_t \sim \mathrm{Bin}\left(N_t, \frac{w(s)}{w(t)}\right), \quad B_s \mid (N_t, N_s, B_t) \sim \mathrm{Hyp}(N_t, B_t, N_s), \quad D_s = N_s - B_s, \quad (8)$$

and $d_t = x_t - x_0$, N_t *and* B_t *are given by equation 6 and* $M_t \sim$ *Bes*$(|d_t|; \Lambda_+(t), \Lambda_-(t))$.

We visualize this process in Fig. 1 where we show the trajectories for the one- and two-step models along with the core composition property that drives bridge models. This setup enables training and sampling from a Count Bridge, see Algorithms 1 and 2. These results leverage our custom CUDA kernel implementing (Devroye, 2002) to enable sampling the Bessel distribution at scale.

In Fig. 1 we can also see that as d_1 grows the slack concentrates at zero. This further connects Count and Schrödinger Bridges Léonard (2013). Let $P^{\kappa}_{\mathrm{ref}}$ be the joint induced by the birth-death kernel. Then, iterating on Count Bridge would yield π^{κ}, similarly to Shi et al. (2023), where we have

$$\pi^{\kappa} \in \arg\min_{\Pi \in \Pi(P_0, P_1)} \mathrm{KL}(\Pi \,\|\, P^{\kappa}_{\mathrm{ref}}), \quad \mathrm{KL}(\Pi \,\|\, P^{\kappa}_{\mathrm{ref}}) = C_{\kappa} + \log\left(\tfrac{2}{\kappa}\right)\mathbb{E}_{\Pi}|X_1 - X_0| - H(\Pi) + o_{\kappa}(1),$$

so $\kappa \downarrow 0$ gives the discrete OT (with cost $|x_1 - x_0|$), while $\kappa \uparrow \infty$ yields $P_0 \otimes P_1$ (see App. A.3). Returning to our setup in Section 2, if $X_1 = X_0 + B_{\sigma}$ (i.e., $\alpha_t \equiv 1,\ \sigma_t \equiv \sigma$), the same projection gives

$$\mathrm{KL}(\Pi \,\|\, P^{\sigma}_{\mathrm{ref}}) = \tilde{C}_{\sigma} + \tfrac{1}{2\sigma^2}\mathbb{E}_{\Pi}\|X_1 - X_0\|^2 - H(\Pi) + o_{\sigma}(1),$$

$\sigma \downarrow 0$ gives the quadratic OT, while $\sigma \uparrow \infty$ yields $P_0 \otimes P_1$. Thus κ echoes entropy regularization in σ.

3.2 Distributional scoring loss for the denoiser

Training requires a distributional loss due to the discrete nature of the space. As shown by Holderrieth et al. (2024), the ELBO for discrete generators (e.g., jump processes) is distributional and cannot be reduced to expectations over point estimates. This mirrors the need for cross-entropy in discrete diffusion and flow models. However, since our data lives on a structured count space, we can go beyond cross-entropy by using a proper scoring rule that better respects the geometry.

Formally, let (X_0, X_t) denote a pair from the forward bridge law at time $t \in [0, 1]$, and let $q_{\theta}(\cdot \mid x_t, t, z)$ be our denoiser. We train q_{θ} using a strictly proper distributional scoring rule (Gneiting & Raftery, 2007; De Bortoli et al., 2025). Fix a negative-type semimetric ρ on $\mathbb{Z}^D$ (e.g. $\rho(x, x') = \|x - x'\|_2^{\beta}$ with $\beta \in (0, 2)$). For any distribution p and outcome y, the energy score is

$$S_{\rho}(p, y) = \tfrac{1}{2}\,\mathbb{E}_{X, X' \sim p}\big[\rho(X, X')\big] - \mathbb{E}_{X \sim p}\big[\rho(X, y)\big],$$

which is strictly proper when ρ is characteristic. The objective integrates the conditional score over the bridge with a user-chosen weight schedule w_t:

$$\mathcal{L}(\theta) = -\int_0^1 w_t \mathbb{E}_{(X_0, X_t)}\Big[S_{\rho}\big(q_{\theta}(\cdot \mid X_t, t, z),\, X_0\big)\Big]dt.$$

We employ the standard U-statistic estimator (Gneiting & Raftery, 2007; De Bortoli et al., 2025).

4 Deconvolution with count bridges

We extend Count Bridges to handle unit–level generation when we only observe aggregates. Consider G units in the one-dimensional case where the group-level state at time t is a vector $\mathbf{X}_t \in \mathbb{Z}^G$ with entries X_{gt} for unit g at time t. Each entry evolves independently according to the bridge in Section 3. The key challenge: we observe the unit–level endpoint $\mathbf{x}_1$ but only the aggregate at time 0, $a_0 = \sum_{g=1}^{G} \mathbf{x}_{g0} \in \mathbb{Z}$, not the unit–level vector $\mathbf{x}_0$. Our goal is to learn a count bridge $q_{\theta}(\mathbf{x}_0 \mid \mathbf{x}_t, t, z)$ that generates unit–level endpoints given start data at time $t = 1$ and side information z.

We formulate this as a generalized EM problem, similar to Rozet et al. (2024), where $\mathbf{X}_0$ is latent and $a_0 = \sum_g \mathbf{X}_{g0}$ is observed. Let $A : \mathbb{Z}^G \to \mathbb{Z}$ be a linear aggregate map (e.g., sums across units, block sums). For (x_t, t, z), the denoiser $q_{\theta}(\cdot \mid x_t, t, z)$ induces a predictive *aggregate* law

$$Q_{\theta}(a \mid x_t, t, z) = \mathbb{P}\big(A(X_0) = a \big| X_t{=}x_t, t, z, X_0 \sim q_{\theta}(\cdot \mid x_t, t, z)\big).$$

In the E-step we will generate "latent" $x_0^{\approx}$ using the model and in the M-step we will use these $x_0^{\approx}$ to train the model at the aggregate level. We summarize the overall procedure in Algorithms 3 and 4.

Require: dataset (x_0, x_1), $w(\cdot), \Lambda_\pm(\cdot)$
for each minibatch **do**
  sample $(x_0, x_1) \sim p_{\text{data}}$
  $t \sim \text{Unif}[0,1]$
  $d_1 \leftarrow x_1 - x_0$
  $M_1 \sim \text{Bes}(|d_1|;\, \Lambda_+(1), \Lambda_-(1))$
  $N_1 \leftarrow |d_1| + 2M_1$
  $B_1 \leftarrow \frac{1}{2}(N_1 + d_1)$
  $N_t \sim \text{Bin}\big(N_1,\, w(t)\big)$
  $B_t \sim \text{Hyp}\big(N_1,\, B_1,\, N_t\big)$
  $x_t \leftarrow x_1 - 2(B_1 - B_t) + (N_1 - N_t)$
  update θ on $-\log q_\theta(x_0 \mid x_t, t)$
end for

Algorithm 1: Training Poisson–BD Bridge

Require: $x_{t_K} = x_1$, model q_θ, $w(\cdot), \Lambda_\pm(\cdot)$
for $k = K, K-1, \ldots, 1$ **do**
  sample $\hat{x}_0 \sim q_\theta(\cdot \mid x_{t_k}, t_k)$
  $d_{t_k} \leftarrow x_{t_k} - \hat{x}_0$
  $M_{t_k} \sim \text{Bes}(|d_{t_k}|;\, \Lambda_+(t_k), \Lambda_-(t_k))$
  $N_{t_k} \leftarrow |d_{t_k}| + 2M_{t_k}$
  $B_{t_k} \leftarrow \frac{1}{2}(N_{t_k} + d_{t_k})$
  $\rho \leftarrow w(t_{k-1})/w(t_k)$
  $N_{t_{k-1}} \sim \text{Bin}\big(N_{t_k},\, \rho\big)$
  $B_{t_{k-1}} \sim \text{Hyp}\big(N_{t_k},\, B_{t_k},\, N_{t_{k-1}}\big)$
  $x_{t_{k-1}} \leftarrow x_{t_k} - 2(B_{t_k} - B_{t_{k-1}}) + (N_{t_k} - N_{t_{k-1}})$
end for
return x_{t_0}

Algorithm 2: Sampling Poisson–BD Bridge

E-Step The ideal E–step would sample from the exact aggregate–conditional law

$$\mathbf{X}_0^\star \sim Q_\theta(\cdot \mid A(\mathbf{X}_0) = a_0, x_t, t, z)\,.$$

We could then use the sampled $\mathbf{x}_0^\star$ as latent variables to sample x_t by running the forward bridge between $(\mathbf{x}_0^\star, \mathbf{x}_1)$ using the unit–level kernel Prop. 3.1.[1] Unfortunately, Q_θ is generally intractable to sample from, given just a unit-level model, so we approximate it through the diffusion sampling process itself. Starting from $\mathbf{x}_1$, we run the sampling process as in Algorithm 2, but at each timestep t_k we: (1) predict $\hat{\mathbf{x}}_0 \sim q_\theta(\cdot \mid \mathbf{x}_{t_k}, t_k, z)$, (2) project $\hat{\mathbf{x}}_0$ to satisfy the aggregate constraint (see Sec. 4), yielding $\tilde{\mathbf{x}}_0$, and (3) perform the sampling step using $\tilde{\mathbf{x}}_0$ as the predicted endpoint. This projection–guided diffusion ensures the aggregate constraint is incorporated throughout the denoising trajectory (see Alg. 3). This process produces latent $x_0^{\approx}$ samples that are consistent with the aggregate constraints, which we can then use in the M-step to train the model. We outline this in App. B.4 and prove that, when learning from aggregates is possible, the EM approach will learn the bridge.

M-Step With these unit-level samples in hand, the M–step runs the bridge process as in Section 3. But instead of computing the loss on the unit-level latents, we compute the loss with respect to the aggregates. Given the ground-truth aggregate a_0, we lift the same strictly proper score to aggregates:

$$S_\rho^{\text{agg}}\big(Q_\theta(\cdot \mid x_t, t, z),\, a_0\big) = \tfrac{1}{2}\,\mathbb{E}\big[\rho(A(X), A(X'))\big] - \mathbb{E}\big[\rho(A(X), a_0)\big],$$

where $X, X' \overset{\text{i.i.d.}}{\sim} q_\theta(\cdot \mid x_t, t, z)$. The training objective from aggregate supervision is then

$$\mathcal{L}_{\text{agg}}(\theta) = -\int_0^1 w_t \mathbb{E}_{(A_0, X_t)}\Big[S_\rho^{\text{agg}}\big(Q_\theta(\cdot \mid X_t, t, z),\, A_0\big)\Big]\, dt$$

with the Monte-Carlo plug-in obtained by sampling $\hat{x}^{(j)} \sim q_\theta$ and forming $\hat{a}^{(j)} = A(\hat{x}^{(j)})$.

Approximate Sampling from the conditional distribution Given a predicted endpoint $\hat{\mathbf{x}}_0$ from our diffusion model and target aggregate a_0, we need to sample from the conditional distribution $Q_\theta(\cdot \mid A(\mathbf{X}_{g0}) = a_0)$. While this is intractable, we can derive a principled approximation.

Proposition 4.1. *Taylor expanding the true conditional law, under certain assumptions on p_{data}, the first-order approximation to the conditional distribution is the KL projection*

$$\Pi(x_0) = \arg\min_{y_0:\, A(y_0) = a_0} D_{\text{KL}}(y_0 \| x_0) = a_0 \cdot \frac{x_{g0}}{\sum_{g'} x_{g'0}}.$$

The proposition shows that the natural rescaling operation is not ad hoc, but is justified as a kind of first-order Taylor approximation to the true conditional distribution (see Appendix B.1). When unit-level training data exist, we can learn a projection $\Pi_\psi(\hat{\mathbf{x}}_0, z, a_0)$ that actually enables sampling conditional on the mean. See Sec. 6 where we show outline how to learn such a projetion.

[1] The same method described here can be used with distributional diffusion on continuous space, but we focus on counts since most often when we observe aggregates we believe they are based on discrete underlying data.

Require: $(\mathbf{x}_1, a_0, z)$, $w(\cdot), \Lambda_{\pm}(\cdot)$, q_θ, Π
1: **for** t from 1 down to τ with steps **do**
2: $\quad \tilde{\mathbf{x}}_0 \sim \Pi(q_\theta(\cdot \mid \mathbf{x}_t, t, z), z, a_0)$
3: $\quad$ Update x_t by running reverse step
4: $\quad\quad$ using steps 4–10 of Alg. 2
5: **end for**
6: $\tilde{\mathbf{x}}_0 \sim \Pi(q_\theta(\cdot \mid \mathbf{x}_t, t, z), z, a_0)$
7: **return** $\tilde{\mathbf{x}}_0$

Algorithm 3: Guided Sampling to Time τ

Require: $(\mathbf{x}_1, a_0, z)$, $w(\cdot), \Lambda_{\pm}(\cdot)$, q_θ, Π
1: **for** each minibatch **do**
2: $\quad$ **E-step:** Sample from $\mathbf{x}_1$ to τ via Alg. 3
3: $\quad$ **M-step:** $t \sim \mathrm{Unif}[0, 1]$
4: $\quad\quad$ Sample x_t via forward bridge on
5: $\quad\quad\quad x_0^{\approx}, x_1$ using steps 4–10 of Alg. 1
6: $\quad$ Update θ on $-\log Q_\theta(a_0 \mid \mathbf{x}_t, t, z)$
7: **end for**

Algorithm 4: Training with Aggregate Supervision

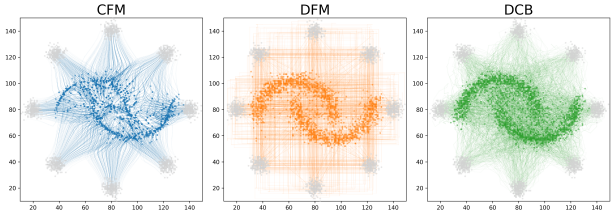


Figure 2: A scaled and rounded variant of the classic 8 gaussian to two moons task. Here we compare the trajectories of continuous flow matching, discrete flow matching, and count bridges. CB achieves the lowest W_2, MMD, and EMD, see Table 5.

5 Related works

Stochastic interpolants. Our formulation allows us to transport any integer-valued distribution p_1 to another integer-valued distribution p_0. In the case of Euclidean state space early works such as (De Bortoli et al., 2021; Vargas et al., 2021; Chen et al., 2021) have shown how to perform such an interpolation leveraging (Entropic) Optimal transport and the concept of Schrödinger Bridges. In more recent works, ignoring the Optimal Transport constraints, several works have proposed to bridge distributions in a more relaxed formulation leveraging the concept of Markov projection, see Peluchetti (2023); Albergo et al. (2023); Delbracio & Milanfar (2023); Liu et al. (2022; 2023) for instance. Those frameworks can be shown to be strictly equivalent to diffusion models in the case where one of the end distribution is a unit Gaussian, see Gao et al. (2025). However, those works are limited to the Euclidean setting, and extension to the integer-valued setting is required.

Discrete diffusion models. Recently, with the advent of language diffusion models such as Ye et al. (2025); Song et al. (2025); Sahoo et al. (2024); Shi et al. (2024); Ou et al. (2024a); Arriola et al. (2025); Nie et al. (2024); Zheng et al. (2023), discrete diffusion models have gained considerable traction. Most works rely on discrete equivalents of the original formulation of diffusion models, explicitly or implicitly replacing the continuous Gaussian noising process by a Continuous-Time Markov Chain (CTMC) Austin et al. (2021); Campbell et al. (2022); Lou et al. (2023); Campbell et al. (2024); Kitouni et al. (2024); Sun et al. (2023). Other approaches include relying on some Euclidean relaxation Chen et al. (2022) or modelling the space of probability Avdeyev et al. (2023); Stark et al. (2024). Similarly, flow matching techniques have been extended to cover this paradigm Gat et al. (2024). Most of these works focus on *categorical* data and therefore consider uninformed forward process such as uniform or masking process. In contrast, in this work, we focus on ordinal data. To the best of our knowledge, the only existing work that also deals with such a process is Blackout Diffusion Santos et al. (2023), which considers a pure-death process where an image is taken to the all-zero limit, as opposed to an endpoint conditioned bridge. Our approach generalizes this setup in two ways: first, we allow births and deaths at every time, recovering their pure birth construction in the limit as $\kappa \to 0$; second, we generalize the process to a bridge which can transport X_1 to X_0.

Finally, we highlight that diffusion models have been extended to the very general setting where only an *infinitesimal generator* is available Benton et al. (2024); Holderrieth et al. (2024). While our work can be seen as an instanciation of this general framework, these general frameworks do not give any information regarding the design of the forward process for integer-valued data, the specific parameretization in terms of slack variables and the necessity of the distributional diffusion loss.

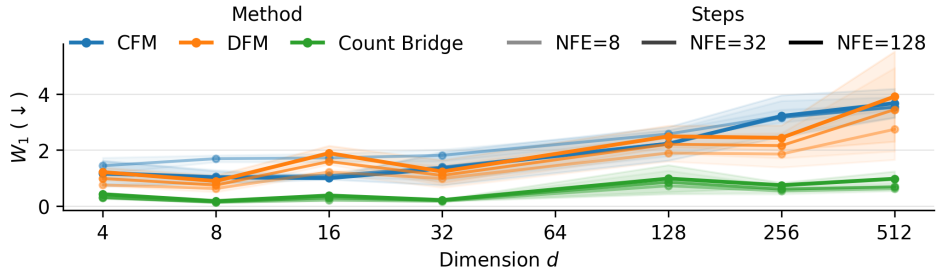


Figure 3: CFM, DFM, and CB on our low-rank mixture of Gaussians transport experiment across dimensions and NFE. See App. C.2 for full details.

Distributional Diffusion Models. In De Bortoli et al. (2025); Shen et al. (2025), the authors learn the conditional distribution $p_{0|t}(x_0|x_t)$ through the use of scoring rules, going beyond the classical training framework of diffusion, which approximates the conditional mean $\mathbb{E}[X_0|X_t = x_t]$. The importance of approximating the covariance was already noted by Nichol & Dhariwal (2021) and further analyzed in (Ho et al., 2020; Nichol & Dhariwal, 2021; Bao et al., 2022a;b; Ou et al., 2024b). In a similar flavor (Xiao et al., 2022) uses a GAN to approximate $p_{0|t}(x_0|x_t)$.

Sequence-to-expression models An ambitious goal in biology is to predict gene expression from DNA sequence information. There have been several attempts to train deep learning models for sequence-to-expression prediction tasks (Barbadilla-Martínez et al., 2025), including Enformer (Avsec et al., 2021), a state-of-the-art transformer-based DNA sequence model. While powerful, Enformer, like the vast majority of sequence-to-expression models, was trained on bulk gene expression data and is not able to predict single-cell expression profiles, missing the cellular heterogeneity and fine-grained regulatory patterns that shape tissue function.

Spatial transcriptomic deconvolution Spatial transcriptomics encompasses a family of recently developed techniques which measure gene expression and spatial location in tissues. The majority of these techniques are not capable of resolving individual cells, instead providing aggregate information over small neighborhoods consisting of on the order of tens of cells (Moses & Pachter, 2022). To address this limitation, a number of deconvolution methods have been developed to infer single-cell level information Li et al. (2023b). The majority of these methods, including cell2location (Kleshchevnikov et al., 2022) and RCTD (Cable et al., 2022), require a paired non-spatially resolved scRNA-seq atlas, and output cluster-level mixture proportions rather than single cell counts. The ideal deconvolution would recover full single-cell count profiles directly from spatial data without requiring external reference atlases. STDeconvolve Miller et al. (2022)

6 Applications

6.1 Synthetic Distributions

Here, we we benchmark count bridges (CB) against continuous flow matching (CFM) (Lipman et al., 2022) and discrete flow matching (DFM) (Gat et al., 2024) across a range of synthetic experiments.

Discrete 8-Gaussians to 2-Moons. We adapt this classic task to the integers. We plot the learned trajectories in Fig 2. Qualitatively CB achieves the best performance. DFM is much more competitive in this experiment than CFM, but DFM trajectories are decoupled from the underlying geometry, whereas CB produces OT-like trajectories similar to CFM. These qualitative evaluations are confirmed quantitatively: CB achieves the best performance across W_2, Energy, and MMD (see App. C.1).

Scaling in Low-Rank Gaussian Mixtures. To test scalability to higher dimensions, we construct integer-valued datasets with fixed intrinsic dimensionality while ambient dimension d increases in powers of two from 4 to 512. Each dataset is a 5-component Gaussian mixture with latent rank $r = 3$, projected to $\mathbb{Z}^d$. In Fig. 3 see that CB has the best scaling in dimensionality (see App. C.2 for more).

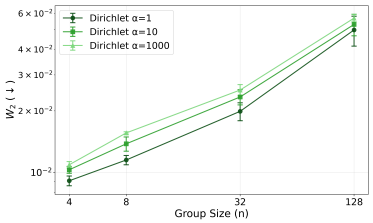


Figure 4: Deconvolution of the low-rank Gaussian mixture.

Deconvolution of Gaussian Mixtures. We extend the low-rank mixture task to evaluate deconvolution capabilities. In this experiment, each observation is an aggregate constructed by summing a group of G samples. For each group, the G samples are drawn from a group-specific Gaussian mixture whose component weights are sampled from a Dirichlet distribution with concentration parameters $(\alpha_1, \ldots, \alpha_5)$. The labels of the G source components are provided as unit-level side information. We then vary the size of the group G and the extent of variation between groups by changing the concentration parameter α (see Appendix C.3 for details). In Fig. 4 we see performance degrades as groups become more uniform and larger. We explore the theoretical limits to deconvolution in Apps. B.4.3 and B.3, which confirm that deconvolution requires between-group heterogeneity to enable identification, which is inherently lost as groups become large. Despite these limits, we demonstrate practical deconvolution on moderately-sized groups in our spatial transcriptomics application (Section 6.3).

6.2 Modelling gene expression at single-cell and single-nucleotide resolution

A central goal in biology is to understand the relationship between DNA sequence and gene expression. Many models have been developed to relate sequence and expression, the most prominent of which, such as Enformer (Avsec et al., 2021) are Transformer-based models which predict expression from sequence. These models are typically trained on bulk data, where gene expression counts are very high such that continuous approximation is effective. More recent work has explored fine-tuning Enformer on single cell data (Hingerl et al., 2024). However, single cell data has significantly lower counts, which may make continuous approximation less suitable.

Here, we use Count Bridges to jointly model sequence and expression counts in single-cell RNA sequencing (scRNA-seq) data. We show two key results. First, we show that Count Bridges enable cell-type specific sequence-to-expression prediction which outperforms Enformer. Second, we show that the aggregate-level conditioning enables CB to deconvolve bulk gene expression profiles into inferred single-cell gene expression profiles.

We model PBMC scRNA-seq counts at the nucleotide level from 10^6 cells and 10^3 donors Yazar et al. (2022). For each nucleotide, we condition on the noisy count x_t, the diffusion time t, a cell-type embedding, and i.i.d. noise z, together with a local genomic context encoded by Enformer. The concatenated features pass through residual multi–head attention blocks and a final head with softplus output. Additionally we train using a small *projection* module Π_ψ—an attention block that jointly processes the nucleotide-level embeddings across positions—to produce a projected $\tilde{x}_0 = \Pi_\psi(\hat{x}_0, a_0, x_t)$ so that when an aggregate is observed we can approximately sample from the conditional $X_0 \mid A(X_0){=}a_0, X_t, t$.

Cell-type specific gene expression We first evaluate the ability of our model to predict expression from sequence, conditional on cell type. As a baseline, we use an Enformer model which is fine-tuned directly on the PBMC dataset. We find that Count Bridge predictions outperform fine-tuned Enformer (Table 1, for results by cell type and further details see App. E).

Bulk deconvolution Count Bridges enable us to lift our unit-level model to deconvolution tasks: given an aggregate of counts across a set of cells, we can make predictions about the single cell profiles by conditioning on the aggregate. We next evaluate the ability of Count Bridges to deconvolve mixtures of cell types from held-out individuals. As a baseline, we compare to CIBERSORTx (Newman et al., 2019), a bulk RNA-seq deconvolution method which operates at the level of genes (rather than single nucleotides), and outputs cell type proportions (not count profiles). To compare we aggregate our nucleotide-level predictions into gene counts and assign each of our deconvolved cells to the closest cell type. Count Bridges achieve better performance on JSD and RMSE (Table 2).

6.3 Deconvolving spatial transcriptomic spots into single-cell counts

Next, we show how count bridges can be used to infer single cell gene expression profiles from spot-level aggregates in spatial transcriptomic data. In spatial transcriptomic data generated by

Method	Bulk MSE	Cell-type MSE
Fine-tuned Enformer	2.590	3.142
Count Bridge	0.601	1.41

Table 1: Sequence-to-expression prediction error

Method	JSD	RMSE
CIBERSORTx	0.193	0.665
Count Bridge	0.166	0.547

Table 2: Cell-type deconvolution error

Visium (Ståhl et al., 2016), it is common to have access to side information beyond the spot-level count aggregates. In particular, many datasets include images of the cells with a nuclear stain (Palla et al., 2022). Count bridges provide a natural way to leverage this cell-level side information to deconvolve aggregate count data. Following the notation in Sec. 4, spot-level counts can be treated as linear aggregates a_0, and single-cell images can be treated as unit-level side information z. We leverage a UViT (Bao et al., 2023) extended to incorporate count and noise patches (see App. D).

To evaluate deconvolution from aggregates, we use a MERFISH mouse brain dataset (Vizgen, 2021) which is resolved at the single-cell level, and artifically aggregate neighborhoods of cells to simulate spot-level Visium data. This synthetic dataset gives us access to spot-level aggregates and their corresponding single cell ground truth, as well as single-cell nuclear images. With this benchmarking setup, we compare our approach to STDeconvolve (Miller et al., 2022), a widely used spatial transcriptomic deconvolution method which is state-of-the-art among reference-free approaches Li et al. (2023b) (see Appendix D for comparisons to reference-based methods). STDeconvolve outputs cell type (cluster identity) proportions for each spot rather than single cell counts. As such, we evalute the quality of deconvolution as the error (root mean squared error and Jensen-Shannon Divergence) of the predicted cell type proportions from true cell type proportions per spot.

Method	RMSE	JSD
STDeconvolve	2.746	0.289
Count Bridge	**0.404**	**0.250**

Table 3: Cell-type error

For count bridges, which provide single-cell count profile predictions rather than cell type proportions, we assign each predicted count profile to a cell type using a nearest neighbor classifier in order to compare against STDeconvolve. Count Bridges outperforms STDeconvolve on both the JSD and the RMSE (Table 3).

Table 4: Count profile error

Comparison	MMD	W_2	Energy
Spot mean	0.409	0.030	41.717
Count Bridge	**0.258**	**0.020**	**15.787**

We next evaluate the quality of the count profiles inferred by Count Bridges. Here, because STDeconvolve does not provide these predictions, we instead consider a simple baseline: predicting the spot-level mean (a_0/G) for each cell. This baseline, while seemingly naive, is actually biologically well-motivated. In spatial transcriptomics, cells within a spot represent local tissue organization where neighboring cells coordinate their functions (Armingol et al., 2021). As such, we expect cells in spatial neighborhoods to have correlated gene expression profiles, making the spot mean a reasonable baseline. Nonetheless, Count Bridges outperform the spot-level mean baseline (see Table 4), showing that Count Bridges can learn meaningful unit-level distributions from real-world aggregate data.

7 Conclusion.

Count Bridges offer a tractable, discrete-native alternative to continuous diffusion models, unifying direct count generation with deconvolution from aggregates. Exact birth–death conditionals on $\mathbb{Z}^d$ let us train and sample without approximation, and we observe order-of-magnitude gains over both continuous and discrete flow matching. We then demonstrate the power of count bridges for nucleotide-level deconvolution of bulk RNA-seq and spatial transcriptomic deconvolution.

Limitations. (i) When counts are well-approximated as continuous, Euclidean models may match or exceed performance. (ii) Identifiability in pure deconvolution degrades as group sizes grow or between-group heterogeneity shrinks, so our EM procedure is most reliable at moderate aggregation. (iii) The projection step we use is a first-order surrogate and lacks serious theoretical support.

Despite these caveats, Count Bridges lay the groundwork for rigorous discrete generative modeling and invite future work on deeper understanding of the projection-guided sampler, sharper identifiability bounds, and generally stronger guarantees for projection-guided EM.

Ethics Statement. This study uses publicly released, de-identified single-cell and spatial transcriptomics datasets under their respective licenses; no new human subject data were collected, and institutional review board (IRB) approval was therefore not required. We do not foresee serious ethical implications to Count Bridges beyond the risks already posed by standard diffusion/flow matching models. Our deconvolution methods could possibly pose some additional privacy risks, but we believe the additional risk is low. We used LLMs to help draft portions of the code used in our experiments and to edit portions of this manuscript. All our models are intended for research use only, not clinical use. LLMs were not used in any way significantly outside the current norms of academic research.

Reproducibility Statement. We have taken significant steps to ensure that all results presented in this work are reproducible. An anonymous source code repository is provided here, containing complete implementations of the Count Bridge framework, including model architectures, training procedures, projection-based deconvolution, and evaluation pipelines. The appendix includes full mathematical derivations and proofs of all theoretical claims. We also provide descriptions of all data preprocessing steps for synthetic benchmarks, PBMC sequence-to-expression prediction, and spatial transcriptomic aggregation, as well as architectural and hyperparameter specifications. Together, these materials are intended to allow independent researchers to fully reproduce our theoretical and empirical findings.

REFERENCES

Nist digital library of mathematical functions. `https://dlmf.nist.gov/`, 2025. See §10.41(ii). Accessed 2025-09-24.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Erick Armingol, Adam Officer, Olivier Harismendy, and Nathan E Lewis. Deciphering cell-cell interactions and communication from gene expression. *Nature reviews. Genetics*, 22(2):71–88, February 2021. ISSN 1471-0056,1471-0064. doi: 10.1038/s41576-020-00292-x.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, pp. 1276–1301. PMLR, 2023.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18 (10):1196–1203, 4 October 2021. ISSN 1548-7091,1548-7105. doi: 10.1038/s41592-021-01252-x.

Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *International Conference on Machine Learning*, 2022a.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022b.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023.

Lucía Barbadilla-Martínez, Noud Klaassen, Bas van Steensel, and Jeroen de Ridder. Predicting gene expression from DNA sequence using deep learning models. *Nature reviews. Genetics*, 26(10): 666–680, 13 May 2025. ISSN 1471-0056,1471-0064. doi: 10.1038/s41576-025-00841-2.

Sean C Bendall, Erin F Simonds, Peng Qiu, El-Ad D Amir, Peter O Krutzik, Rachel Finck, Robert V Bruggner, Rachel Melamed, Angelica Trejo, Olga I Ornatsky, Robert S Balderas, Sylvia K Plevritis, Karen Sachs, Dana Pe'er, Scott D Tanner, and Garry P Nolan. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science (New York, N.Y.)*, 332(6030):687–696, 6 May 2011. ISSN 0036-8075,1095-9203. doi: 10.1126/science.1198704.

Joe Benton, Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. From denoising diffusions to denoising markov models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(2):286–301, 2024.

Dylan M Cable, Evan Murray, Luli S Zou, Aleksandrina Goeva, Evan Z Macosko, Fei Chen, and Rafael A Irizarry. Robust decomposition of cell type mixtures in spatial transcriptomics. *Nature biotechnology*, 40(4):517–526, April 2022. ISSN 1087-0156,1546-1696. doi: 10.1038/s41587-021-00830-w.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

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. *arXiv preprint arXiv:2402.04997*, 2024.

Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of schr\" odinger bridge using forward-backward sdes theory. *arXiv preprint arXiv:2110.11291*, 2021.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. *arXiv preprint arXiv:2502.02483*, 2025.

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *arXiv preprint arXiv:2303.11435*, 2023.

Luc Devroye. Simulating bessel random variables. *Statistics & probability letters*, 57(3):249–257, 2002.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Johannes C Hingerl, Laura D Martens, Alexander Karollus, Trevor Manz, Jason D Buenrostro, Fabian J Theis, and Julien Gagneur. scooby: Modeling multi-modal genomic profiles from DNA sequence at single-cell resolution. *bioRxiv.org: the preprint server for biology*, pp. 2024.09.19.613754, 23 September 2024. doi: 10.1101/2024.09.19.613754.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. *arXiv preprint arXiv:2410.20587*, 2024.

Ouail Kitouni, Niklas Nolte, James Hensman, and Bhaskar Mitra. Disk: A diffusion model for structured knowledge, 2024. URL https://arxiv.org/abs/2312.05253.

Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 21 May 2015. ISSN 0092-8674,1097-4172. doi: 10.1016/j.cell.2015.04.044.

Vitalii Kleshchevnikov, Artem Shmatko, Emma Dann, Alexander Aivazidis, Hamish W King, Tong Li, Rasa Elmentaite, Artem Lomakin, Veronika Kedlian, Adam Gayoso, Mika Sarkin Jain, Jun Sung Park, Lauma Ramona, Elizabeth Tuck, Anna Arutyunyan, Roser Vento-Tormo, Moritz Gerstung, Louisa James, Oliver Stegle, and Omer Ali Bayraktar. Cell2location maps fine-grained cell types in spatial transcriptomics. *Nature biotechnology*, 40(5):661–671, 13 May 2022. ISSN 1087-0156,1546-1696. doi: 10.1038/s41587-021-01139-4.

Christian Léonard. A survey of the schr\" odinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Haoyang Li, Juexiao Zhou, Zhongxiao Li, Siyuan Chen, Xingyu Liao, Bin Zhang, Ruochi Zhang, Yu Wang, Shiwei Sun, and Xin Gao. A comprehensive benchmarking with practical guidelines for cellular deconvolution of spatial transcriptomics. *Nature Communications*, 14:1548, March 2023a. ISSN 2041-1723. doi: 10.1038/s41467-023-37168-7.

Haoyang Li, Juexiao Zhou, Zhongxiao Li, Siyuan Chen, Xingyu Liao, Bin Zhang, Ruochi Zhang, Yu Wang, Shiwei Sun, and Xin Gao. A comprehensive benchmarking with practical guidelines for cellular deconvolution of spatial transcriptomics. *Nature communications*, 14(1):1548, 21 March 2023b. ISSN 2041-1723,2041-1723. doi: 10.1038/s41467-023-37168-7.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I 2 sb: Image-to-image schr\" odinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Brendan F Miller, Feiyang Huang, Lyla Atta, Arpan Sahoo, and Jean Fan. Reference-free cell type deconvolution of multi-cellular pixel-resolution spatially resolved transcriptomics data. *Nature communications*, 13(1):2339, 29 April 2022. ISSN 2041-1723,2041-1723. doi: 10.1038/s41467-022-30033-z.

Lambda Moses and Lior Pachter. Museum of spatial transcriptomics. *Nature methods*, 19(5):534–546, 10 May 2022. ISSN 1548-7091,1548-7105. doi: 10.1038/s41592-022-01409-2.

Aaron M Newman, Chloé B Steen, Chih Long Liu, Andrew J Gentles, Aadel A Chaudhuri, Florian Scherer, Michael S Khodadoust, Mohammad S Esfahani, Bogdan A Luca, David Steiner, et al. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature biotechnology*, 37(7):773–782, 2019.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. *arXiv preprint arXiv:2410.18514*, 2024.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024a.

Zijing Ou, Mingtian Zhang, Andi Zhang, Tim Z Xiao, Yingzhen Li, and David Barber. Diffusion model with optimal covariance matching. *arXiv preprint arXiv:2406.10808*, 2024b.

Giovanni Palla, Hannah Spitzer, Michal Klein, David Fischer, Anna Christina Schaar, Louis Benedikt Kuemmerle, Sergei Rybakov, Ignacio L Ibarra, Olle Holmberg, Isaac Virshup, Mohammad Lotfollahi, Sabrina Richter, and Fabian J Theis. Squidpy: a scalable framework for spatial omics analysis. *Nature methods*, 19(2):171–178, 28 February 2022. ISSN 1548-7091,1548-7105. doi: 10.1038/s41592-021-01358-2.

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

A Raj, P V D van den Bogaard, Scott A Rifkin, A van Oudenaarden, and Sanjay Tyagi. Imaging individual mRNA molecules using multiple singly labeled probes. *Nature methods*, 5(10):877–879, 21 September 2008. ISSN 1548-7091,1548-7105. doi: 10.1038/nmeth.1253.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. *Advances in Neural Information Processing Systems*, 37:87647–87682, 2024.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Javier E Santos, Zachary R Fox, Nicholas Lubbers, and Yen Ting Lin. Blackout diffusion: generative diffusion models in discrete-state spaces. In *International Conference on Machine Learning*, pp. 9034–9059. PMLR, 2023.

Xinwei Shen, Nicolai Meinshausen, and Tong Zhang. Reverse markov learning: Multi-step generative models for complex distributions. *arXiv preprint arXiv:2502.13747*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36:62183–62223, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang, Lin Yan, Xiaoying Jia, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Yonghui Wu, and Hao Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference, 2025. URL https://arxiv.org/abs/2508.02193.

Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, Annelie Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlén, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisén. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science (New York, N.Y.)*, 353(6294):78–82, 1 July 2016. ISSN 0036-8075,1095-9203. doi: 10.1126/science.aaf2403.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models, 2023. URL https://arxiv.org/abs/2211.16750.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

Vizgen. Vizgen Data Release V1.0, May 2021. Title of the publication associated with this dataset: Mouse Brain Receptor Map.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022.

Seyhan Yazar, Jose Alquicira-Hernandez, Kristof Wing, Anne Senabouth, M Grace Gordon, Stacey Andersen, Qinyi Lu, Antonia Rowson, Thomas R P Taylor, Linda Clarke, Katia Maccora, Christine Chen, Anthony L Cook, Chun Jimmie Ye, Kirsten A Fairfax, Alex W Hewitt, and Joseph E Powell. Single-cell eQTL mapping identifies cell type-specific genetic control of autoimmune disease. *Science (New York, N.Y.)*, 376(6589):eabf3041, 8 April 2022. ISSN 0036-8075,1095-9203. doi: 10.1126/science.abf3041.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

A Count Bridges

A.1 Poisson Birth–Death Bridge on $\mathbb{Z}$

We start by showing where the Binomial and Hypergeoemtric distributions emerge in our framework. Then we will prove that if we condition on the amount of slack these distributions compose. Finally we will show that using the Bessel slack law we have composition while mixing over the slack distribution.

A.1.1 Sampling process

Let $\lambda_\pm$ be Borel rates with cumulants $\Lambda_\pm(t) = \int_0^t \lambda_\pm(s)\,ds$ and set $w(t) = \frac{\Lambda_+(t)+\Lambda_-(t)}{\Lambda_+(1)+\Lambda_-(1)}$. Draw independent Poisson processes $(B_t)_{t\in[0,1]} \sim \mathrm{Poi}(\Lambda_+)$ and $D_{t\in[0,1]} \sim \mathrm{Poi}(\Lambda_-)$, and define $X_t = X_0 + B_t - D_t$. Condition on endpoints $X_0 = x_0$, $X_1 = x_1$ and write $d_1 := x_1 - x_0$, $N_1 := B_1 + D_1$, $B_1 = \frac{1}{2}(N_1 + d_1)$.

Let $N_t := \#\{s \le t : \Delta X_s \neq 0\}$ be the total jump count up to t and let (overloading notation) B_t also denote the number of $+1$ jumps among those N_t. Under the superposed non-homogenous Poisson process with rate $\lambda_+ + \lambda_-$, the N_1 unordered jump times are i.i.d. with cdf $w(\cdot)$, and, conditional on (N_1, B_1), the $+1$ labels form a uniformly random subset of size B_1 among $\{1, \dots, N_1\}$, independent of times. Therefore,

$$N_t \mid N_1 \sim \mathrm{Bin}\big(N_1,\, w(t)\big), \qquad B_t \mid (N_1, B_1, N_t) \sim \mathrm{Hyp}\big(N_1,\, B_1,\, N_t\big).$$

More generally, for $0 < s < t < 1$ with $\rho = w(s)/w(t)$,

$$N_s \mid N_t \sim \mathrm{Bin}\big(N_t,\, \rho\big), \qquad B_s \mid (N_t, B_t, N_s) \sim \mathrm{Hyp}\big(N_t,\, B_t,\, N_s\big).$$

A.1.2 Composition Conditional on M_1

Now we will prove composition while conditioning on the slack M_1. Fix integers $x_0, x_1 \in \mathbb{Z}$ with displacement $d_1 := x_1 - x_0$. Fix the slack variable $M_1 \in \mathbb{N}$. We require two elementary composition lemmas; the first is classical, the second is a counting exercise.

Lemma A.1 (Binomial composition). *Let $K \sim \mathrm{Bin}(N, p)$ and, conditional on K, let $L \sim \mathrm{Bin}(K, q)$. Then $L \sim \mathrm{Bin}(N, pq)$ and (K, L) has joint pmf*

$$\mathbb{P}\{K = k,\, L = \ell\} = \binom{N}{\ell}(pq)^{\ell}(1-pq)^{N-\ell}\binom{N-\ell}{k-\ell}\Big(\frac{p(1-q)}{1-pq}\Big)^{k-\ell}\Big(\frac{1-p}{1-pq}\Big)^{N-k}.$$

Proof. First, write down the joint probability

$$\mathbb{P}(K = k,\, L = \ell) = P(K = k)\; P(L = \ell \mid K = k) = \binom{N}{k}p^{k}(1-p)^{N-k}\;\binom{k}{\ell}q^{\ell}(1-q)^{\,k-\ell}.$$

Using the combinatorial identity $\binom{N}{k}\binom{k}{\ell} = \binom{N}{\ell}\binom{N-\ell}{k-\ell}$, we get

$$\mathbb{P}(K = k,\, L = \ell) = \binom{N}{\ell}(pq)^{\ell}(1-pq)^{N-\ell}\;\binom{N-\ell}{k-\ell}\Big(\tfrac{p(1-q)}{1-pq}\Big)^{k-\ell}\Big(\tfrac{1-p}{1-pq}\Big)^{N-k}.$$

To show that L is $\mathrm{Bin}(N, pq)$, we explicitly marginalize over all possible $k \geq \ell$:

$$\mathbb{P}(L = \ell) = \sum_{k=\ell}^{N}\mathbb{P}(K = k,\, L = \ell) = \sum_{k=\ell}^{N}\binom{N}{\ell}(pq)^{\ell}(1-pq)^{N-\ell}\binom{N-\ell}{k-\ell}\Big(\tfrac{p(1-q)}{1-pq}\Big)^{k-\ell}\Big(\tfrac{1-p}{1-pq}\Big)^{N-k}.$$

Factor out everything not depending on k, and change variables $m = k - \ell$:

$$\mathbb{P}(L = \ell) = \binom{N}{\ell}(pq)^{\ell}(1-pq)^{N-\ell}\sum_{m=0}^{N-\ell}\binom{N-\ell}{m}\Big(\tfrac{p(1-q)}{1-pq}\Big)^{m}\Big(\tfrac{1-p}{1-pq}\Big)^{(N-\ell)-m}.$$

But the sum is exactly $\sum_{m=0}^{N-\ell}\binom{N-\ell}{m}a^{m}b^{(N-\ell)-m} = (a+b)^{N-\ell}$ with

$$a = \frac{p(1-q)}{1-pq}, \quad b = \frac{1-p}{1-pq}, \qquad a + b = 1.$$

Hence that sum is 1, and we conclude

$$\mathbb{P}(L = \ell) = \binom{N}{\ell}(pq)^{\ell}(1-pq)^{N-\ell},$$

i.e. $L \sim \mathrm{Bin}(N, pq)$. □

Lemma A.2 (Nested hypergeometric draws). *Fix an urn with N balls of which B are white. Draw $N_t = k$ balls without replacement and record $B_t = b$ whites. From the k drawn, draw $N_s = j, (\leq k)$ and record $B_s = a$ whites. Then the marginal law obtained by the two-step procedure satisfies*

$$\mathbb{P}\{N_s = j,\, B_s = a \mid N, B\} = \binom{N}{j}^{-1}\binom{B}{a}\binom{N-B}{j-a},$$

which equals the pmf of the single hypergeometric draw $\mathrm{Hyp}(N, B, j)$.

Proof. Observe that the two–stage procedure:

1. pick a subset of size k from $\{1, \ldots, N\}$,
2. then pick a subset of size j from that first subset

is exactly equivalent to "pick a subset of size j directly from $\{1, \ldots, N\}$," because any j-subset can be realized in some order of two draws and all orders are equally likely under simple "draw-without-replacement."

Therefore, conditional on the final sample size $N_s = j$, each of the $\binom{N}{j}$ subsets of size j from the N items is equally likely. Among those, exactly

$$\binom{B}{a}\binom{N-B}{j-a}$$

subsets contain exactly a of the B "success" items. Hence

$$\mathbb{P}\big(B_s = a \;\big|\; N_s = j\big) = \frac{\binom{B}{a}\binom{N-B}{j-a}}{\binom{N}{j}},$$

which is precisely $\mathrm{Hyp}(N, B, j)$. □

Theorem A.3 (Consistency). *For $0 < s < t < 1$ the two-stage sampling rule $(1 \to t \to s)$ yields exactly the same law for (N_s, B_s) as the single-stage rule $(1 \to s)$. Explicitly*

$$N_s \mid N_1 \sim \mathrm{Bin}\big(N_1,\, w(s)\big), \qquad B_s \mid N_s, B_1 \sim \mathrm{Hyp}\big(N_1,\, B_1,\, N_s\big).$$

Proof. Write $p = w(t)$ and $q = w(s)/w(t)$. By Lemma A.1, $N_s \mid N$ obtained via $N \to N_t \to N_s$ is $\mathrm{Bin}(N_1, pq) = \mathrm{Bin}\big(N_1, w(s)\big)$, coinciding with the direct rule.

Conditional on $N_s = j$ and B_1, the colour counts obey Lemma A.2 and thus coincide with a single hypergeometric draw. Hence, the two sampling routes coincide in distribution. □

A.2 Observable Markov property via Poisson thinning and Bessel slack

We now show that the bridge obtained from the Poisson birth–death (BD) reference is *Markov in the observable* X_t (i.e., its one–step kernels satisfy Chapman–Kolmogorov without augmenting the state) if and only if, at each time, we mix over *Bessel slack* (Skellam–conditioned) endpoint counts. The proof uses only classical Poisson thinning/superposition and the form of the Skellam conditioning.

Throughout, let $\alpha(t) = \Lambda_+(t)$, $\beta(t) = \Lambda_-(t)$ and $w(t) = \frac{\alpha(t)+\beta(t)}{\alpha(1)+\beta(1)} \in [0, 1]$. Write $d_t := X_t - X_0$, $d := d_1 = X_1 - X_0$.

A.2.1 Closure under thinning and difference conditioning

Lemma A.4 (Thinning closure). *If $B_1 \sim \mathrm{Poi}(\alpha)$ and $D_1 \sim \mathrm{Poi}(\beta)$ are independent, then for any $t \in (0, 1)$*

$$B_t \mid B_1 \sim \mathrm{Bin}(B_1, w(t)), \quad D_t \mid D_1 \sim \mathrm{Bin}(D_1, w(t)) \quad \textit{independently},$$

and marginally $(B_t, D_t) \sim \mathrm{Poi}\,(\alpha_t) \times \mathrm{Poi}\,(\beta_t)$ with $\alpha_t = w(t)\alpha,\ \beta_t = w(t)\beta$.

Lemma A.5 (Bessel slack). *Condition on the difference $d_t = B_t - D_t$ at any fixed t. Then the conditional law of the endpoint counts $(B_t, D_t) \mid \{B_t - D_t = d_t\}$ is supported on the lattice $\{(m+|d_t|, m) : m \in \mathbb{N}\}$ (if $d_t \geq 0$, or the swapped lattice if $d_t < 0$) with*

$$\Pr\{M_t = m \mid d_t\} \;\propto\; \frac{(\alpha_t\beta_t)^m}{(m + |d_t|)!\, m!}, \qquad \alpha_t = w(t)\alpha,\ \beta_t = w(t)\beta,$$

i.e. the Bessel slack *pmf. The normalizer is $(\alpha_t\beta_t)^{-|d_t|/2}\, I_{|d_t|}\big(2\sqrt{\alpha_t\beta_t}\big)$.*

Proof sketches. Lemma A.4 is standard Poisson thinning. Lemma A.5 is the usual derivation of the Skellam conditioning: for $d_t \geq 0$, $\Pr(B_t = m+d_t, D_t = m) = e^{-(\alpha_t+\beta_t)}\, \alpha_t^{m+d_t}\beta_t^m/((m+d_t)!\, m!)$, and dividing by $\Pr(B_t - D_t = d_t)$ yields the stated form with Bessel normalizer. □

By Lemma A.5, the posterior over the slack given the *observable* d_t has the same functional form at every t, with parameters simply scaled by thinning:

$$\pi_t(m \mid d_t) \;\propto\; \frac{\big(\alpha_t\beta_t\big)^m}{(m + |d_t|)!\, m!}, \qquad \alpha_t = w(t)\alpha,\ \beta_t = w(t)\beta. \tag{9}$$

Hence $\pi_t(\cdot \mid d_t)$ depends *only* on the current observable state x_t (through d_t) and on t via (α_t, β_t).

A.2.2 OBSERVABLE KERNELS

Fix $0 < u < t < s < 1$. For each $m \in \mathbb{N}$, let $K^{(m)}_{a|b}$ denote the one–step BH transition kernel at times $b \to a$ *conditional on* $M = m$ (equivalently on the endpoint counts/total jumps). By binomial composition and hypergeometric consistency, each $K^{(m)}$ satisfies CK:

$$K^{(m)}_{s|t} \;=\; K^{(m)}_{s|u} * K^{(m)}_{u|t}. \tag{10}$$

Define the *observable* kernel by mixing out the slack with the time-t posterior equation 9:

$$K_{a|b}(x_b, x_0) \;:=\; \sum_{m=0}^{\infty} \pi_b(m \mid d_b)\, K^{(m)}_{a|b}(x_b, x_0), \qquad d_b = x_b - x_0.$$

Theorem A.6 (Observable Markov property and CK)**.** *Under the Poisson BD reference with Bessel slack mixing, the observable kernels* are Markov *and satisfy Chapman–Kolmogorov:*

$$K_{s|t} \;=\; K_{s|u} * K_{u|t} \qquad \textit{for all } 0 < u < t < s < 1.$$

Proof. We prove the CK identity against an arbitrary bounded test function φ. Condition on (X_t, x_0) and write $d_t = X_t - x_0$.

Using the tower property and equation 10,

$$\mathbb{E}\big[\varphi(X_s) \mid X_t, x_0, M\big] = \int \varphi(x)\, K^{(M)}_{s|t}(x; X_t, x_0) = \int \mathbb{E}\big[\varphi(X_s) \mid X_u, x_0, M\big]\, K^{(M)}_{u|t}(\mathrm{d}x_u; X_t, x_0).$$

Average w.r.t. the *time-t* posterior $\pi_t(m \mid d_t)$, valid by equation 9:

$$\mathbb{E}\big[\varphi(X_s) \mid X_t, x_0\big] = \sum_m \pi_t(m \mid d_t) \int \mathbb{E}\big[\varphi(X_s) \mid X_u, x_0, M{=}m\big]\, K^{(m)}_{u|t}(\mathrm{d}x_u; X_t, x_0).$$

Given X_u and x_0, the *time-u* posterior over M is again equation 9 with $(\alpha_u, \beta_u) = (w(u)\alpha, w(u)\beta)$ and $d_u = X_u - x_0$, by Lemmas A.4–A.5. Therefore

$$\sum_m \pi_t(m \mid d_t)\, K^{(m)}_{u|t}(\mathrm{d}x_u; X_t, x_0) \;=\; K_{u|t}(\mathrm{d}x_u; X_t, x_0),$$

and likewise $\sum_m \pi_u(m \mid d_u)\, K^{(m)}_{s|u} = K_{s|u}$.

Combine the previous displays to obtain

$$\begin{aligned}\mathbb{E}\big[\varphi(X_s) \mid X_t, x_0\big] &= \int \Big(\sum_m \pi_u(m \mid d_u)\, \mathbb{E}\big[\varphi(X_s) \mid X_u, x_0, M{=}m\big]\Big)\, K_{u|t}(\mathrm{d}x_u; X_t, x_0) \\ &= \int \mathbb{E}\big[\varphi(X_s) \mid X_u, x_0\big]\, K_{u|t}(\mathrm{d}x_u; X_t, x_0),\end{aligned}$$

which is the CK identity $K_{s|t} = K_{s|u} * K_{u|t}$ in weak form. Since φ is arbitrary, the kernel identity holds. □

The core intuition here is that the only statistic of X_u relevant for the slack posterior is $d_u = X_u - X_0$, and its likelihood under any $M = m$ is Skellam with parameters $(\alpha_u, \beta_u) = (w(u)\alpha, w(u)\beta)$. Thus the posterior family over M is *closed* under time changes (thinning) and depends only on the current observable state. This is precisely the "lumpability" needed for Markovianity in X_t.

A.3 LINK WITH SCHRÖDINGER BRIDGE

We have that the distribution of $X_t | X_0, X_1$ is the bridge distribution of a process such that for any t

Set

$$X_t = X_0 + B_t - D_t.$$

Then $X_1 \mid X_0$ is Skellam with pmf

$$K^{\kappa}_{1|0}(x_1 \mid x_0) = \exp[-\Lambda_+(1) - \Lambda_-(1)] \left(\frac{\Lambda_+(1)}{\Lambda_-(1)}\right)^{\frac{x_1 - x_0}{2}} I_{|x_1 - x_0|}(2\kappa).$$

Let $P^{\kappa}_{\mathrm{ref}} := P_0 \otimes K^{\kappa}_{1|0}$ be the *reference joint*. For any coupling $\Pi \in \Pi(P_0, P_1)$,

$$\begin{aligned}\mathrm{KL}\big(\Pi \,\|\, P^{\kappa}_{\mathrm{ref}}\big) &= -H(\Pi) - \mathbb{E}_{\Pi}\big[\log K^{\kappa}_{1|0}(X_1 \mid X_0)\big] + C_{\kappa}(P_0) \\ &= C_{\kappa}(P_0, P_1) - \mathbb{E}_{\Pi}\big[\log I_{|X_1 - X_0|}(2\kappa)\big] - H(\Pi),\end{aligned}$$

where $C_{\kappa}(P_0, P_1)$ collects terms depending only on the marginals (including $\Lambda_{\pm}(1)$ and $\mathbb{E}[X_1 - X_0]$).

Using the facts that $I_{\nu}(z) = \frac{e^z}{\sqrt{2\pi z}}\,(1 + o(1))$ as $z \to \infty$ and $I_{\nu}(z) = (z/2)^{\nu}/\Gamma(\nu+1)\,(1 + o(1))$ as $z \to 0^+$ (NIS, 2025, §10.41(ii)), we obtain

$$\mathrm{KL}\big(\Pi \,\|\, P^{\kappa}_{\mathrm{ref}}\big) = \begin{cases} C - H(\Pi) + o_{\kappa}(1), & \kappa \to \infty, \\ C + \log\left(\frac{2}{\kappa}\right) \mathbb{E}_{\Pi}|X_1 - X_0| - H(\Pi) + o_{\kappa}(1), & \kappa \to 0^+, \end{cases}$$

which yields the conclusions in the main text: high noise leads to the independent coupling $P_0 \otimes P_1$; low noise gives discrete OT with cost $\mathbb{E}_{\Pi}|X_1 - X_0|$.

B Lifting Count Bridges to Aggregates

We make two central assumptions that enable deconvolution.

Assumption B.1 (Realizability and recoverability)**.** *There exists $\theta^\star$ such that for all $t \in [0, 1]$:*

1. ***Realizability:*** *$Q_{\theta^\star}(a_0 \mid \mathbf{x}_t, t, z) = p_{data}(a_0 \mid \mathbf{x}_t, t, z)$ almost surely*
2. ***Recoverability:*** *The aggregate-to-unit map has local modulus $\kappa_{loc}(t)$: for θ near $\theta^\star$,*
   $D_{KL}(p_{data}(\mathbf{x}_0 \mid \mathbf{x}_t, t, z) \,\|\, q_\theta(\cdot \mid \mathbf{x}_t, t, z)) \le \kappa_{loc}(t) \cdot D_{KL}(p_{data}(a_0 \mid \mathbf{x}_t, t, z) \,\|\, Q_\theta(\cdot \mid \mathbf{x}_t, t, z))$

Recoverability means that the aggregate distribution uniquely determines the unit-level distribution—if we know the sum perfectly, we can deduce the summands. This is not always possible. Consider the simplest case: if $X_1, X_2 \sim \text{Poisson}(\lambda_1), \text{Poisson}(\lambda_2)$ are independent, their sum is $\text{Poisson}(\lambda_1 + \lambda_2)$. Observing only the sum, we cannot distinguish $(\lambda_1 = 3, \lambda_2 = 2)$ from $(\lambda_1 = 4, \lambda_2 = 1)$—both yield $\text{Poisson}(5)$. Now if we had side information that identified the "component" each Poisson was drawn from we could identify this, but it illustrates the difficulties we will face here.

Recoverability holds when units have sufficient diversity. Formally, it requires that units are conditionally independent given $(\mathbf{x}_1, z)$ and have distinct factorial cumulant signatures—essentially, different statistical fingerprints that survive aggregation. For count data, this means units must have different parameters (e.g., different Poisson rates or negative binomial dispersions) that are distinguishable through the covariates z.

Theorem B.10 in Appendix B.4.3 provides precise conditions: when the factorial cumulant generating functions $C_{X_{g^0}}(t) = F(t; \psi_g)$ form an identifiable system and covariates provide sufficient labeling to distinguish units. In practice, this means units should have heterogeneous characteristics captured by z—for example, different images associated with the transciptomics is spatial single cell data.

Recoverability faces fundamental limits as the number of units G grows. By the central limit theorem, when $G \to \infty$, the standardized aggregate $(A_0 - \mu_G)/\sigma_G$ converges to a Gaussian regardless of the unit-level distributions. The higher-order cumulants that distinguish different unit configurations vanish at rate $O(G^{-k/2})$ for order $k > 2$, leaving only the mean and variance (Appendix B.3).

This CLT collapse means our method is most powerful for moderate G (tens to hundreds of units) where unit heterogeneity is preserved in aggregates. For very large G, additional structure is needed—either parametric constraints (e.g., unit parameters follow a low-dimensional model), multiple aggregate observations under different conditions, or direct observation of some unit-level data.

We illustrate these issues in Fig. 4 in the main text: as the dirichlet concentration increases the group-level mixture weights concentrate. When combined with large group sizes this forces all groups to become identical. This gives a clear empirical sense for the limits of deconvolution.

B.1 Approximately Sampling Conditional on the Sum

The most central part of our deconvolution approach is our projection operation. Here we sketch a formal characterization of this operation and justify it based on a Taylor expansion around the true conditional distribution.

We first give a completely formal statement of our theorem:

Theorem B.2 (Rescaling emerges from first-order conditional)**.** *Let p_{data} be the prior law of X_0 and write the aggregate $A_0 = A(X_{g0})$ with $\mu = \mathbb{E}_{P_\theta}[A_0]$ and $\Sigma = \mathrm{Cov}_{P_\theta}(A_0)$ (finite, p.d.). For a target aggregate a_0, set $\delta := a_0 - \mu$. Then the aggregate-conditional law $Q_\theta(\cdot \mid A_0 = a_0)$ admits the Radon–Nikodym form*

$$\frac{dQ_\theta}{dP_\theta}(x_0) = \exp\Big\{\lambda^\top \textstyle\sum_g x_{g0} - A(\lambda)\Big\}, \qquad \lambda = \Sigma^{-1}\delta + O(\|\delta\|^2),$$

where $A(\lambda) = \log \mathbb{E}_{P_\theta}\big[e^{\lambda^\top \sum_g X_{g0}}\big]$. The KL projection forms a first-order approximation:

$$T^\star(x_0) = \arg\min_{y_0:\, \sum_g y_{g0} = a_0} D_{\mathrm{KL}}(y_0 \| x_0),$$

which for non-overlapping groups yields the simple scaling update

$$T^\star(x_0)_g^{(d)} \;=\; x_{g0}^{(d)} \cdot \frac{a_0^{(d)}}{\sum_{g'} x_{g'0}^{(d)}}\,, \qquad d = 1, \ldots, D.$$

We prove Theorem B.2 (Section 3.2) under explicit regularity, giving a first-order expansion for the tilt parameter and $O(\|\delta\|^2)$ control of the KL gap.

Assumption B.3 (Cramér regularity and nondegenerate covariance)**.** *Let P_θ be the prior law of X_0 and $A_0 = \sum_g X_{g0} \in \mathbb{Z}^D_{\geq 0}$. The cumulant generating function*

$$A(\lambda) \;:=\; \log \mathbb{E}_{P_\theta}\big[e^{\lambda^\top A_0}\big]$$

exists and is finite on an open neighborhood $\mathcal{N}$ of $\lambda = 0$, is twice continuously differentiable on $\mathcal{N}$, and $\Sigma := \nabla^2 A(0) = \mathrm{Cov}_{P_\theta}(A_0)$ is positive definite. We also assume $\nabla^2 A$ is locally Lipschitz on a smaller neighborhood $\mathcal{N}_0 \subset \mathcal{N}$.

Definition B.4 (Exponential tilt and I-projection)**.** *For $\lambda \in \mathcal{N}$, define the tilted law on X_0:*

$$\frac{dP_{\theta,\lambda}}{dP_\theta}(x_0) \;=\; \exp\big\{\lambda^\top A_0(x_0) - A(\lambda)\big\}.$$

Let $a_0 \in \mathbb{R}^D$ be a feasible aggregate and $\delta := a_0 - \mu$ with $\mu := \nabla A(0) = \mathbb{E}_{P_\theta}[A_0]$. The I-projection *of P_θ onto the affine moment set $\{Q : \mathbb{E}_Q[A_0] = a_0\}$ is $P_{\theta,\hat\lambda}$ with $\hat\lambda$ the unique solution of*

$$\nabla A(\hat\lambda) = a_0.$$

Lemma B.5 (Duality and uniqueness)**.** *Under Assumption B.3, the map $\lambda \mapsto \nabla A(\lambda)$ is a local diffeomorphism at 0, hence for $\|a_0 - \mu\|$ sufficiently small there exists a unique $\hat\lambda \in \mathcal{N}_0$ such that $\nabla A(\hat\lambda) = a_0$. Moreover,*

$$\mathrm{KL}\Big(P_{\theta,\hat\lambda} \,\Big\|\, P_\theta\Big) = A(\hat\lambda) - \hat\lambda^\top a_0 \quad \textit{and} \quad \mathbb{E}_{P_{\theta,\hat\lambda}}[A_0] = a_0.$$

Proof. $\nabla^2 A(0) = \Sigma \succ 0$ implies invertibility of the Jacobian at 0. The inverse function theorem yields a local inverse ψ of ∇A near μ, with $\hat\lambda = \psi(a_0)$. The KL identity is standard for exponential families; the moment identity is by construction. □

Lemma B.6 (First-order expansion of the dual parameter). *Let L be a Lipschitz constant for $\nabla^2 A$ on $\mathcal{N}_0$. Then, for δ small enough,*

$$\hat{\lambda} \;=\; \Sigma^{-1}\delta \;+\; r(\delta), \qquad \|r(\delta)\| \;\le\; \tfrac{L}{2}\,\|\Sigma^{-1}\|^2\,\|\delta\|^2.$$

Proof. Taylor expand ∇A at 0: $\nabla A(\lambda) = \mu + \Sigma\lambda + R(\lambda)$ with $\|R(\lambda)\| \le \frac{L}{2}\|\lambda\|^2$. Solve $\mu + \Sigma\hat{\lambda} + R(\hat{\lambda}) = \mu + \delta$ to obtain $\hat{\lambda} = \Sigma^{-1}(\delta - R(\hat{\lambda}))$, hence the bound. □

Lemma B.7 (KL and expectation errors). *As $\delta \to 0$,*

$$\mathrm{KL}\Big(P_{\theta,\hat{\lambda}} \,\Big\|\, P_\theta\Big) \;=\; \tfrac{1}{2}\,\delta^\top\Sigma^{-1}\delta + O(\|\delta\|^3), \qquad \big\|\mathbb{E}_{P_{\theta,\hat{\lambda}}}[f(X_0)] - \mathbb{E}_{Q_\theta(\cdot\mid A_0=a_0)}[f(X_0)]\big\| \;=\; O(\|\delta\|^2),$$

for any f with $\mathbb{E}_{P_\theta}[|f(X_0)|] < \infty$ whenever the exact conditional $Q_\theta(\cdot \mid A_0 = a_0)$ exists.

Proof. Expand $A(\hat{\lambda})$ to second order using Lemma B.6 and standard cumulant properties. For expectations, note that both $P_{\theta,\hat{\lambda}}$ and $Q_\theta(\cdot \mid A_0 = a_0)$ are I-projections onto the same affine moment set; the latter is the exact conditional when it exists. Bregman (KL) Pythagorean identities give that their KL gap is $O(\|\delta\|^2)$, which implies the stated expectation difference by Pinsker and boundedness-by-integrability. □

Theorem B.8 (Proof of Theorem B.2). *Under Assumption B.3, for $\delta = a_0 - \mu$ small,*

$$\frac{dQ_\theta}{dP_\theta}(x_0) = \exp\big\{\hat{\lambda}^\top A_0(x_0) - A(\hat{\lambda})\big\}, \qquad \hat{\lambda} = \Sigma^{-1}\delta + O(\|\delta\|^2),$$

and $\mathrm{KL}(Q_\theta \| P_{\theta,\hat{\lambda}}) = O(\|\delta\|^2)$. For per-coordinate column constraints, the I-projection is the multiplicative scaling

$$T^\star(x_0)_g^{(d)} = x_{g0}^{(d)} \cdot \frac{a_0^{(d)}}{\sum_{g'} x_{g'0}^{(d)}}, \qquad d = 1, \ldots, D.$$

Proof. Combine Lemmas B.5–B.7. The explicit scaling is the closed-form I-projection onto the linear constraints $\sum_g y_{g0}^{(d)} = a_0^{(d)}$ with KL geometry ("IPF step"); it follows from separability across d. □

B.2 Conditions for Realizability and Recoverability

We provide concrete conditions under which the key assumptions of recoverability and realizability hold for count data.

B.2.1 Factorial cumulant framework

For nonnegative integer-valued X, define the factorial moment generating function (FMGF):

$$M_X(t) = \mathbb{E}[(1+t)^X], \quad t \in \mathbb{R} \text{ near } 0,$$

and the factorial cumulant generating function (FCGF):

$$C_X(t) = \log M_X(t) = \sum_{k\ge 1} \frac{\kappa_k(X)}{k!} t^k,$$

where $\kappa_k(X)$ are the factorial cumulants.

Lemma B.9 (Properties of factorial cumulants). *(a) If X, Y are independent nonnegative integer-valued, then $C_{X+Y}(t) = C_X(t) + C_Y(t)$.*

(b) If $\{f_\psi : \psi \in \Psi\}$ are real-analytic near 0 with injective $\psi \mapsto f_\psi$, and $\sum_{g=1}^G f_{\psi_g} \equiv \sum_{g=1}^G f_{\psi'_g}$, then the multisets $\{\psi_g\}$ and $\{\psi'_g\}$ coincide.

B.2.2 SUFFICIENT CONDITIONS FOR RECOVERABILITY

Theorem B.10 (Recoverability via factorial cumulants)**.** *Assume:*

1. *Conditionally on $\mathcal{C} = \sigma(\mathbf{X}_1, Z)$, the units $(X_{10}, \ldots, X_{G0})$ are independent.*
2. *Each unit's FCGF has the form $C_{X_{g0}}(t) = F(t; \psi_g)$ where $F(\cdot; \psi)$ is real-analytic near $t = 0$ and $\psi \mapsto F(\cdot; \psi)$ is injective.*
3. *The covariates provide labeling: distinct units with distinct ψ_g have distinct labels $\lambda_g = \lambda(\mathbf{X}_1, Z, g)$ almost surely.*

Then the aggregate map $\mathcal{A}_\mathcal{C}$ is injective, ensuring recoverability.

Proof. By independence and Lemma B.9(a), $C_{A_0}(t) = \sum_{g=1}^{G} F(t; \psi_g)$. If two specifications yield the same aggregate law, their FCGF sums agree. By Lemma B.9(b), the multisets coincide. The labeling removes permutation ambiguity, yielding $\psi_g = \psi'_g$ for each g. □

B.3 LARGE-G LIMITS: CLT-INDUCED NON-IDENTIFIABILITY

We show that even when recoverability holds for finite G, the deconvolution problem can become ill-posed as $G \to \infty$ due to central limit phenomena.

B.3.1 THE FUNDAMENTAL TENSION

As G grows, a fundamental statistical phenomenon emerges: the aggregate distribution converges to a Gaussian regardless of the specific unit-level distributions, losing the fine-grained information needed for deconvolution.

B.3.2 CLT COLLAPSE OF AGGREGATE INFORMATION

Theorem B.11 (Loss of identifiability under CLT scaling)**.** *Let $\{X_{g0}^{(G)} : 1 \le g \le G\}$ be conditionally independent given $\mathcal{C} = \sigma(\mathbf{X}_1, Z)$, with*

$$\mu_G = \sum_{g=1}^{G} \mathbb{E}[X_{g0}^{(G)} \mid \mathcal{C}], \quad \sigma_G^2 = \sum_{g=1}^{G} \mathit{Var}(X_{g0}^{(G)} \mid \mathcal{C}) < \infty.$$

Under Lindeberg conditions, any two sequences of unit-level distributions that produce the same (μ_G, σ_G^2) yield asymptotically identical aggregate distributions:

$$\frac{A_0^{(G)} - \mu_G}{\sigma_G} \Rightarrow \mathcal{N}(0, 1) \quad \textit{as } G \to \infty.$$

Proof. The Lindeberg-Feller CLT applies to both sequences. Since they share the same first two moments, their standardized aggregates converge to the same Gaussian limit, making them indistinguishable through aggregate observations. □

B.3.3 IMPLICATIONS FOR FACTORIAL CUMULANTS

Recall from Section B.4.3 that recoverability relies on the factorial cumulants $\kappa_k(A_0) = \sum_g \kappa_k(X_{g0})$ determining the individual unit parameters. Under CLT scaling:

Corollary B.12 (Vanishing higher-order cumulants)**.** *For standardized aggregates, the k-th factorial cumulant scales as $O(\sigma_G^{-k+2})$ for $k \ge 3$. Thus:*

$$\frac{\kappa_k(A_0^{(G)})}{\sigma_G^k} \to 0 \quad \textit{as } G \to \infty, \quad k \ge 3.$$

Only the first two cumulants (mean and variance) survive in the limit.

This means the factorial cumulant signature that enables recoverability for finite G becomes asymptotically uninformative—all unit-level configurations with the same total mean and variance are indistinguishable.

B.4 Gradient Bounds for Deconvolution

A subtle issue arises in the theoretical analysis of the EM algorithm we present in the main text: if we always generate training pairs by running the full reverse trajectory from $\mathbf{x}_1$ to $\mathbf{x}_0$, errors in early predictions can compound. Each reverse step conditions on the previous prediction, so mistakes propagate and potentially amplify. The model could learn from its own errors, leading to distribution shift.

We resolve this through a simple but crucial observation: at time $t = 1$ (the noisy endpoint), we are training on the actual data, so as long as we can learn unit-level distributions from aggregates this can serve as a "backstop" from a theoretical point of view.

Our modified training strategy exploits this guarantee while allowing the model to benefit from iterative refinement when possible. At each epoch, we evaluate the aggregate prediction quality at different reverse trajectory endpoints $\tau \in \{t_1 = 1, t_2, \ldots, t_K\}$. For each minibatch, we run Alg. 2 (note, not the guided sampling) and compute the aggregate score for the sampled $\hat{x}_{0,t}$ for each timepoint and use the lowest scoring timepoint to form our "latent" $x_0^{\approx}$. We cannot use the guided sampling since it is guaranteed to match the aggregates at late times. But once we choose the end time we can then sample using Alg. 3 stopping at time τ^* and projecting to incorporate the aggregate constraints.

If there is significant compounding error, the score will be worse for smaller τ (longer trajectories), and the procedure naturally falls back to $\tau = 1$ where convergence is guaranteed. However, when the model is well-calibrated, earlier times often achieve better scores because they benefit from multiple rounds of refinement.

Now we establish that training from aggregates with adaptive end-time selection produces gradients that approximate the oracle unit-level gradients, with the approximation quality determined by the best aggregate prediction achievable. This proof is essentially straightforward: we assume that we can learn from aggregates using the model at $t = 1$ and then show that our EM procedure will not go wrong given this backstop.

Empirically, we often find that using the full trajectory ($\tau = 0$) works well without explicit adaptation, suggesting the projection guidance effectively prevents severe compounding. An important area for future work is developing a more satisfying theory that explains this strong performance.

Assumption B.13 (Bounded Fisher information)**.** *The aggregate Fisher information satisfies* $\sup_{(\mathbf{x}_t,t,z)} tr\, I_\theta(\mathbf{x}_t, t, z) \le C_I < \infty$*, and the unit-level score has bounded second moment* $\mathbb{E}_{q_\theta}[\|\nabla_\theta \log q_\theta(\mathbf{x}_0 \mid \mathbf{x}_t, t, z)\|^2] \le C_{unit}(t)$.

Theorem B.14 (Gradient bounds under adaptive training)**.** *Under Assumptions B.1–B.13, define the aggregate risk at time* τ*:*

$$R_\tau(\theta) = \mathbb{E}[D_{KL}(p_{data}(a_0 \mid \mathbf{x}_\tau, \tau, z) \,\|\, Q_\theta(a_0 \mid \mathbf{x}_\tau, \tau, z))]$$

where $\mathbf{x}_\tau$ *is generated by Algorithm 3. Let* $g_\tau(\theta)$ *be the gradient from aggregate scoring and* $g_\tau^{unit}(\theta)$ *the oracle unit-level gradient. Then:*

$$\|g_\tau(\theta) - g_\tau^{unit}(\theta)\| \le \left(\sqrt{2C_I} + \sqrt{2C_{unit}(\tau)\kappa_{loc}(\tau)}\right)\sqrt{R_\tau(\theta)}.$$

For adaptive selection $\tau^* = \arg\min_\tau R_\tau(\theta)$*, the gradient error is controlled by the minimum achievable risk across all times.*

This theorem reveals the power of adaptive training: the gradient approximation quality depends on the *best* prediction achievable at any time, not a fixed time. When the model is poorly calibrated, $\tau = 1$ minimizes risk (where exact aggregate-conditional sampling is possible). As training progresses, earlier times may achieve lower risk through iterative refinement, automatically improving gradient quality. The proof uses the Fisher identity and Pinsker's inequality to bound the gradient gap in terms of the aggregate KL divergence.

Combined with the recoverability assumption, this ensures that minimizing aggregate risk leads to learning the correct unit-level model. Once correct at any time, the θ-free property of the bridge kernel guarantees correct distributions at all times, enabling consistent training across different trajectory endpoints.

Throughout, expectations are taken under the population law. For any time $\tau \in [0,1]$, the state $\mathbf{x}_\tau$ is generated *recursively* by Algorithm 3 (reverse sampling with projection Π used only to draw $\tilde{\mathbf{x}}_0$ as an internal step). The *loss is always pre-projection*: we score $Q_\theta(a_0 \mid \mathbf{x}_\tau, \tau, z)$, never Π.

Write the aggregate score function

$$\ell_\theta(a_0; \mathbf{x}_\tau, \tau, z) \;:=\; -\log Q_\theta(a_0 \mid \mathbf{x}_\tau, \tau, z), \quad s_\theta(a_0 \mid \mathbf{x}_\tau, \tau, z) \;:=\; \nabla_\theta \log Q_\theta(a_0 \mid \mathbf{x}_\tau, \tau, z)$$

so that the population *aggregate* gradient at time τ is

$$g_\tau(\theta) \;:=\; \mathbb{E}\Big[\nabla_\theta \ell_\theta(A_0; \mathbf{x}_\tau, \tau, Z)\Big] \;=\; -\,\mathbb{E}\Big[s_\theta(A_0 \mid \mathbf{x}_\tau, \tau, Z)\Big].$$

Define the *oracle unit-level* score and gradient at time τ

$$u_\theta(\mathbf{x}_0 \mid \mathbf{x}_\tau, \tau, z) \;:=\; \nabla_\theta \log q_\theta(\mathbf{x}_0 \mid \mathbf{x}_\tau, \tau, z), \qquad g_\tau^{\text{unit}}(\theta) \;:=\; \mathbb{E}\Big[-\,u_\theta(\mathbf{X}_0 \mid \mathbf{x}_\tau, \tau, Z)\Big]$$

(the gradient one would take if $\mathbf{X}_0$ were observable).

For the aggregate risk we use the KL divergence

$$R_\tau(\theta) \;:=\; \mathbb{E}\Big[D_{\mathrm{KL}}\Big(p_{\text{data}}(A_0 \mid \mathbf{x}_\tau, \tau, Z) \,\Big\|\, Q_\theta(A_0 \mid \mathbf{x}_\tau, \tau, Z)\Big)\Big].$$

B.4.1 Two elementary lemmas

The first lemma is the (conditional) Fisher identity plus a Cauchy–Schwarz step.

Lemma B.15 (Aggregate score bias bound)**.** *For any fixed* $(\mathbf{x}_\tau, \tau, z)$*,*

$$\Big\| \mathbb{E}\big[s_\theta(A_0 \mid \mathbf{x}_\tau, \tau, z) \,\big|\, \mathbf{x}_\tau, \tau, z\big] \Big\| \;\le\; \sqrt{\operatorname{tr} I_\theta(\mathbf{x}_\tau, \tau, z)} \cdot \sqrt{2\, D_{\mathrm{KL}}\Big(p_{data}(A_0 \mid \mathbf{x}_\tau, \tau, z) \,\Big\|\, Q_\theta(A_0 \mid \mathbf{x}_\tau, \tau, z)\Big)},$$

where $I_\theta(\mathbf{x}_\tau, \tau, z) = \mathrm{Var}_{Q_\theta(\cdot \mid \mathbf{x}_\tau, \tau, z)}[\, s_\theta(\cdot \mid \mathbf{x}_\tau, \tau, z)\,]$.

Proof. Under $Q_\theta(\cdot \mid \mathbf{x}_\tau, \tau, z)$, $\mathbb{E}[s_\theta] = 0$. Let $L = \frac{dp_{\text{data}}}{dQ_\theta}(A_0 \mid \mathbf{x}_\tau, \tau, z)$. Then $\mathbb{E}_{p_{\text{data}}}[s_\theta] = \mathbb{E}_{Q_\theta}[s_\theta(L-1)]$. By Cauchy–Schwarz, $\|\mathbb{E}_{Q_\theta}[s_\theta(L-1)]\| \le \sqrt{\mathbb{E}_{Q_\theta}\|s_\theta\|^2} \cdot \sqrt{\mathbb{E}_{Q_\theta}(L-1)^2}$. The first factor is $\sqrt{\operatorname{tr} I_\theta(\mathbf{x}_\tau, \tau, z)}$. The second equals $\sqrt{\chi^2(p_{\text{data}} \| Q_\theta)} \le \sqrt{2\, D_{\mathrm{KL}}(p_{\text{data}} \| Q_\theta)}$ in the local regime near equality,[2] which yields the claim. □

The second lemma translates aggregate misfit to unit-level misfit via the local modulus.

Lemma B.16 (Unit-level score bias via recoverability)**.** *Fix* $(\mathbf{x}_\tau, \tau, z)$ *and assume the local modulus in Assumption B.1. Then*

$$\Big\| \mathbb{E}\big[u_\theta(\mathbf{X}_0 \mid \mathbf{x}_\tau, \tau, z) \,\big|\, \mathbf{x}_\tau, \tau, z\big] \Big\| \;\le\; \sqrt{2\, C_{unit}(\tau)\, \kappa_{loc}(\tau)} \cdot \sqrt{D_{\mathrm{KL}}\Big(p_{data}(A_0 \mid \mathbf{x}_\tau, \tau, z) \,\Big\|\, Q_\theta(A_0 \mid \mathbf{x}_\tau, \tau, z)\Big)}.$$

Proof. With $p(\cdot) = p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{x}_\tau, \tau, z)$ and $q(\cdot) = q_\theta(\cdot \mid \mathbf{x}_\tau, \tau, z)$, the same step as above gives $\|\mathbb{E}_p[u_\theta]\| \le \sqrt{\mathbb{E}_q\|u_\theta\|^2} \cdot \sqrt{2\, D_{\mathrm{KL}}(p\|q)}$. Bound $\mathbb{E}_q\|u_\theta\|^2 \le C_{\text{unit}}(\tau)$ by Assumption B.13. Then apply the local modulus (Assumption B.1) to replace $D_{\mathrm{KL}}(p\|q)$ by $\kappa_{\text{loc}}(\tau)\, D_{\mathrm{KL}}\big(p_{\text{data}}(A_0 \mid \mathbf{x}_\tau, \tau, z) \| Q_\theta(\cdot \mid \mathbf{x}_\tau, \tau, z)\big)$. □

B.4.2 Proof of Theorem B.14

Proof of Theorem B.14. By definitions,

$$g_\tau(\theta) - g_\tau^{\text{unit}}(\theta) \;=\; -\,\mathbb{E}\Big[s_\theta(A_0 \mid \mathbf{x}_\tau, \tau, Z)\Big] \;+\; \mathbb{E}\Big[u_\theta(\mathbf{X}_0 \mid \mathbf{x}_\tau, \tau, Z)\Big].$$

[2] Locally (when the two distributions are close), $D_{\chi^2} \le 2\, D_{\mathrm{KL}}$; more generally, they are second-order equivalent by standard f-divergence comparisons. This is automatically satisfied in a neighborhood of $\theta^\star$ under Assumption B.1.

Apply the triangle inequality and condition on $(\mathbf{x}_\tau, \tau, Z)$:

$$\|g_\tau(\theta) - g_\tau^{\text{unit}}(\theta)\| \le \mathbb{E}\Big[\, \big\|\mathbb{E}[s_\theta(A_0 \mid \mathbf{x}_\tau, \tau, Z) \mid \mathbf{x}_\tau, \tau, Z]\big\| \,\Big] + \mathbb{E}\Big[\, \big\|\mathbb{E}[u_\theta(\mathbf{X}_0 \mid \mathbf{x}_\tau, \tau, Z) \mid \mathbf{x}_\tau, \tau, Z]\big\| \,\Big].$$

Use Lemma B.15 for the first term and Lemma B.16 for the second, then apply Jensen and Assumptions B.13–B.1:

$$\|g_\tau(\theta) - g_\tau^{\text{unit}}(\theta)\| \;\le\; \Big(\sqrt{2\,C_I} \;+\; \sqrt{2\,C_{\text{unit}}(\tau)\,\kappa_{\text{loc}}(\tau)}\Big)\;\sqrt{R_\tau(\theta)}.$$

Finally, for the adaptive choice $\tau^* \in \arg\min_\tau R_\tau(\theta)$, monotonicity gives $\sqrt{R_{\tau^*}(\theta)} \le \sqrt{R_\tau(\theta)}$ for all τ, so the same bound with $R_{\tau^*}(\theta)$ holds, which is exactly the theorem. □

B.4.3 Remarks on scope and identifiability

Realizability and local modulus. Assumption B.1 asks that aggregate equality implies unit-level equality with a local modulus $\kappa_{\text{loc}}(t)$. Concrete sufficient conditions follow from identifiability of the factorial-cumulant generating family of the units and diversity of covariates z (see Theorem B.10).

Large-G limits. As G grows, higher-order cumulants in the aggregate attenuate (Appendix B.3), so $\kappa_{\text{loc}}(t)$ may deteriorate unless additional structure is imposed (parametric shrinkage across units, multiple aggregates, or occasional unit-level labels).

B.4.4 What adaptive end-time buys you

Defining $R_\tau(\theta)$ using the *recursive* $\mathbf{x}_\tau$ makes the comparison truly training-aligned: your per-example choice $\tau^* = \arg\min_\tau R_\tau(\theta)$ yields the *tightest* bound

$$\|g_{\tau^*}(\theta) - g_{\tau^*}^{\text{unit}}(\theta)\| \;\le\; \big(\sqrt{2\,C_I} + \sqrt{2\,C_{\max}\kappa_{\max}}\big)\,\sqrt{R_{\tau^*}(\theta)},$$

with $C_{\max} = \sup_\tau C_{\text{unit}}(\tau)$ and $\kappa_{\max} = \sup_\tau \kappa_{\text{loc}}(\tau)$ (finite in the realizable neighborhood). Intuitively, as the sampler refines its $\mathbf{x}_\tau$ along the reverse path, whichever time slice permits the *best* aggregate prediction also delivers the *smallest* gap to the oracle unit-level bridge gradient.

C Synthetic distributions

All synthetic tasks use the same base architecture with a 4-layer MLP with 128 dimensional hidden layers. We scale the inputs and outputs in dimension, so for example the DFM and CE-CB have $d \times 256$ dimensional outputs (since we clip all datasets to use a range of 256 to make for easy tokenization). The energy score models take inputs in $d +$ noise_dim and we use noise_dim $= 100$ throughout. We ran all experiments with Adam using both $lr = 1e-3, 2e-4$ and present results for the best performing learning rate for each method. We use a cosine warmup for the learning rate for 100 steps. For all experiments we use gradient norm clipping to size 1, batch size 256, and train for 500 epochs. For the energy score models, we use exponential model averaging, which is crucial to good performance. Full details are available in the codebase.

For the flow matching we use $\sigma = 0.1$ following best practices (we tested larger σ but saw large degradations in performance). For the bessel sampler we use $\sqrt{\Lambda_+\Lambda_-} = 32$.

C.1 Discrete 8-Gaussians to 2-Moons

C.1.1 Dataset

For qualitative evaluation, we adapt the classic continuous "8-Gaussians to 2-Moons" task into a fully discrete, integer-valued setting suitable for count-based flow matching. Each dataset consists of 50,000 paired samples $(x_0, x_1) \in \mathbb{Z}^2$, constructed as follows.

Source distribution (x_0). We generate samples from the standard two-moons dataset in $\mathbb{R}^2$ using `make_moons` with noise level `noise` $= 0.1$. The moons are shifted to be approximately centered at the origin by subtracting $(0.5, 0.25)$.

Target distribution (x_1). We construct an 8-component Gaussian mixture arranged evenly on a circle of radius 2.0 in $\mathbb{R}^2$. Each component has isotropic Gaussian noise with variance matching $\texttt{noise} = 0.1$. A sample is generated by first selecting one of the 8 components uniformly at random, then drawing from the corresponding Gaussian.

Integerization. Both source and target samples are mapped to the integer lattice by

$$x \mapsto \text{round}(\text{clip}(x \cdot \texttt{scale} + \texttt{offset}, \texttt{min\_value}, \texttt{value\_range} - 1)),$$

with parameters $\texttt{scale} = 30.0$, $\texttt{offset} = 80.0$, $\texttt{min\_value} = 0$, and $\texttt{value\_range} = 196$. This procedure ensures that all outputs fall in the discrete vocabulary $\{0, 1, \ldots, 195\}^2$, but the scales are chosen so that essentially no values are actually clipped.

C.1.2 Results

We present a visualization of the learned trajectories in Fig. 2 and the full details in Table 5. Count bridges achieve uniformly the best performance using the distributional losses, that is the cross entropy or energy scores with the energy score uniformly best.

Table 5: Discrete Moons Results: Eight Gaussians → Two Moons

Method	MMD	W_2	Energy
CFM	0.065 ± 0.019	0.049 ± 0.008	0.874 ± 0.246
DFM	0.010 ± 0.002	0.010 ± 0.002	0.035 ± 0.014
Count Bridge (CE)	0.0065 ± 0.0023	0.0080 ± 0.0009	0.026 ± 0.004
Count Bridge (ES)	**0.0044 ± 0.0018**	**0.0052 ± 0.0007**	**0.0098 ± 0.0029**
Count Bridge (MSE)	0.030 ± 0.000	0.033 ± 0.001	0.366 ± 0.015

C.2 Low-Rank Gaussian Mixture

C.2.1 Dataset

For synthetic evaluation, we use a pre-sampled integer-valued Gaussian mixture dataset that scales with the ambient dimension d while fixing the latent rank at $r = 3$. Each dataset consists of $50{,}000$ paired samples $(x_0, x_1) \in \mathbb{Z}^d$ generated according to the following procedure:

Mixture construction. We define a $k = 5$ component Gaussian mixture in latent space $\mathbb{R}^r$ with $r = 3$ (we hold these parameters constant as we scale in d):

- **Means.** Component means are drawn from $\mathcal{N}(0, \sigma^2 I)$ with scale $\sigma = \texttt{mean\_scale}/\sqrt{r}$, and shifted to lie near the center of the integer range. We set $\texttt{mean\_scale} = 20.0$.
- **Covariances.** Each covariance is constructed by sampling eigenvalues from an exponential distribution with scale $\texttt{cov\_scale} = 10.0$, clamped below $\texttt{min\_eigenvalue} = 0.1$, and conjugating by a random orthogonal matrix.
- **Mixture weights.** Weights are drawn from a Dirichlet$(1, \ldots, 1)$ prior, yielding a random simplex vector.

Projection to $\mathbb{R}^d$. Latent samples $z \in \mathbb{R}^3$ are mapped to the ambient space via a random projection matrix $P \in \mathbb{R}^{d \times r}$ with entries scaled by $\texttt{projection\_scale}/\sqrt{r}$, where $\texttt{projection\_scale} = 1.0$. To avoid degeneracy, isotropic Gaussian noise $\epsilon \sim \mathcal{N}(0, \texttt{noise\_scale}^2 I_d)$ with $\texttt{noise\_scale} = 1.0$ is added after projection:

$$y = Pz + \epsilon, \qquad z \sim \text{MoG}_r,\ \epsilon \sim \mathcal{N}(0, I_d).$$

Integerization. Projected samples $y \in \mathbb{R}^d$ are rounded to the nearest integer and reflected into the bounded range $[\texttt{min\_value}, \texttt{value\_range} - 1] = [0, 255]$ to ensure validity of DFM.

Scaling in d. The intrinsic latent structure is fixed at $r = 3$, while the output dimension d is varied across experiments (e.g. $d = 5, 16, 32, 128, 256, 512$). This construction produces datasets with constant intrinsic complexity but increasing ambient dimension, providing a natural test of how models scale in d.

C.2.2 RESULTS

The central scaling results are presented visually in Fig. 3. Here we also present the full experimental details in Table 6.

C.3 DECONVOLUTION GAUSSIAN MIXTURE DATASET

We extend the low-rank Gaussian mixture task (Appendix C.2) to evaluate deconvolution capabilities under controlled conditions. Each observation is formed by aggregating a group of G unit-level samples into a single count vector.

Group construction. For each group, component proportions are drawn from a Dirichlet distribution with concentration parameter α, yielding group-specific mixture weights. The G unit-level samples are then drawn independently from the corresponding mixture components. Both the aggregated sum $X_0 \in \mathbb{Z}^d$ and the individual unit-level labels $z \in \{0, 1\}^{G \times k}$ are retained, enabling evaluation of methods under both aggregate-only and aggregate+unit supervision.

Experimental variation. We vary two factors that control the difficulty of deconvolution:

- **Group size:** $G \in \{4, 8, 32, 128\}$, which determines how many unit-level samples are aggregated. Larger groups yield more uniform averages and less information about component heterogeneity.
- **Dirichlet concentration:** $\alpha \in \{1, 10, 1000\}$, which controls variability in group-specific mixture weights. Small α values produce heterogeneous groups (informative for deconvolution), while large α values yield nearly uniform group proportions (uninformative).

Dataset parameters. We fix the ambient dimension at $d = 4$, latent rank at $r = 3$, number of mixture components $k = 5$, and use the same mixture parameterization as in the low-rank dataset (means scaled by 20.0, covariances scaled by 10.0 with minimum eigenvalue 0.1, projection scale 1.0, isotropic noise 1.0, and integerization into $[0, 255]$). Each dataset contains $5{,}000$ groups, drawn from a base pool of $50{,}000$ pre-sampled mixture samples.

Results. As shown in Fig. 4, deconvolution performance degrades as groups become larger and more uniform. This matches the theoretical results in Appendices B.4.3 and B.3, which establish that deconvolution requires between-group heterogeneity for identification, a property that is inherently lost as G grows. We present detailed results across metrics in Tables 7, 8, 9.

D SPATIAL TRANSCRIPTOMIC DECONVOLUTION

D.1 DATASET

Preprocessing We used the publicly available mouse brain MERFISH dataset from Vizgen (2021). We subset the data to a particular slice (slice 1, replicate 2). We used the transcript puncta and nuclear segmentation masks as provided with the dataset. For gene expression, we used the raw transcript counts without applying standard single-cell preprocessing pipelines. For each cell, we resized the DAPI image to 256x256 pixels by padding.

Aggregation To simulate a Visium-style spatial transcriptomics dataset, we aggregated the single-cell MERFISH data. A grid of spots was defined with a center-to-center distance of $100\mu m$ and a spot radius of $55\mu m$. The gene expression profile for each simulated spot was then generated by summing the transcript counts of all identified cells whose nuclei fell within the circular bounds of that spot.

Table 6: Performance Comparison Across Dimensions and Methods

Dim	Method	NFE	MMD	W_2	EMD
4	CFM	8	0.027 ± 0.014	0.019 ± 0.002	0.716 ± 0.420
		32	0.026 ± 0.017	0.015 ± 0.004	0.584 ± 0.460
		128	0.026 ± 0.018	0.015 ± 0.004	0.565 ± 0.469
	DFM	8	0.025 ± 0.004	0.011 ± 0.001	0.245 ± 0.053
		32	0.035 ± 0.003	0.014 ± 0.001	0.458 ± 0.078
		128	0.046 ± 0.005	0.018 ± 0.002	0.759 ± 0.144
	Count Bridge	8	**0.0053 ± 0.0007**	**0.0040 ± 0.0004**	**0.020 ± 0.004**
		32	**0.0054 ± 0.0010**	**0.0042 ± 0.0002**	**0.023 ± 0.005**
		128	**0.0098 ± 0.0008**	**0.0058 ± 0.0003**	**0.055 ± 0.008**
8	CFM	8	0.041 ± 0.013	0.025 ± 0.004	1.05 ± 0.18
		32	0.039 ± 0.012	0.014 ± 0.004	0.456 ± 0.147
		128	0.040 ± 0.010	0.014 ± 0.003	0.421 ± 0.128
	DFM	8	0.026 ± 0.007	0.011 ± 0.001	0.204 ± 0.067
		32	0.034 ± 0.009	0.011 ± 0.003	0.317 ± 0.142
		128	0.042 ± 0.011	0.012 ± 0.003	0.497 ± 0.228
	Count Bridge	8	**0.0036 ± 0.0007**	**0.0023 ± 0.0001**	**0.0068 ± 0.0024**
		32	**0.0038 ± 0.0011**	**0.0026 ± 0.0006**	**0.0077 ± 0.0028**
		128	**0.0050 ± 0.0015**	**0.0029 ± 0.0003**	**0.012 ± 0.002**
16	CFM	8	0.066 ± 0.011	0.028 ± 0.001	2.08 ± 0.54
		32	0.053 ± 0.011	0.017 ± 0.001	0.788 ± 0.211
		128	0.052 ± 0.011	0.015 ± 0.001	0.647 ± 0.163
	DFM	8	0.078 ± 0.001	0.017 ± 0.000	1.20 ± 0.06
		32	0.100 ± 0.005	0.022 ± 0.002	1.92 ± 0.28
		128	0.118 ± 0.017	0.025 ± 0.004	2.72 ± 0.86
	Count Bridge	8	**0.0067 ± 0.0014**	**0.0035 ± 0.0003**	**0.025 ± 0.007**
		32	**0.011 ± 0.001**	**0.0045 ± 0.0004**	**0.048 ± 0.007**
		128	**0.017 ± 0.001**	**0.0057 ± 0.0004**	**0.090 ± 0.013**
32	CFM	8	0.145 ± 0.024	0.030 ± 0.001	4.63 ± 1.28
		32	0.131 ± 0.043	0.026 ± 0.005	3.14 ± 1.72
		128	0.131 ± 0.052	0.022 ± 0.006	3.09 ± 1.97
	DFM	8	0.079 ± 0.027	0.016 ± 0.008	1.23 ± 0.76
		32	0.089 ± 0.023	0.017 ± 0.006	1.55 ± 0.85
		128	0.100 ± 0.027	0.018 ± 0.007	1.99 ± 1.10
	Count Bridge	8	**0.0083 ± 0.0008**	**0.0024 ± 0.0003**	**0.021 ± 0.002**
		32	**0.010 ± 0.002**	**0.0031 ± 0.0006**	**0.029 ± 0.008**
		128	**0.010 ± 0.002**	**0.0034 ± 0.0007**	**0.034 ± 0.007**
64	CFM	8	0.296 ± 0.077	0.042 ± 0.008	13.43 ± 6.74
		32	0.313 ± 0.099	0.046 ± 0.014	16.07 ± 9.93
		128	0.326 ± 0.107	0.049 ± 0.017	17.94 ± 11.55
	DFM	8	0.105 ± 0.033	0.022 ± 0.007	1.71 ± 1.07
		32	0.126 ± 0.039	0.020 ± 0.005	2.57 ± 1.58
		128	0.147 ± 0.048	0.022 ± 0.006	3.59 ± 2.20
	Count Bridge	8	**0.020 ± 0.004**	**0.0051 ± 0.0005**	**0.072 ± 0.019**
		32	**0.027 ± 0.002**	**0.0061 ± 0.0002**	**0.112 ± 0.014**
		128	**0.029 ± 0.001**	**0.0065 ± 0.0005**	**0.138 ± 0.018**
128	CFM	8	0.335 ± 0.009	0.038 ± 0.003	12.89 ± 1.09
		32	0.276 ± 0.034	0.033 ± 0.008	10.59 ± 3.51
		128	0.260 ± 0.048	0.032 ± 0.008	10.55 ± 4.74
	DFM	8	0.205 ± 0.036	0.042 ± 0.005	6.66 ± 2.47
		32	0.236 ± 0.031	0.043 ± 0.005	9.06 ± 2.63
		128	0.259 ± 0.028	0.050 ± 0.011	10.90 ± 2.92
	Count Bridge	8	**0.128 ± 0.066**	**0.014 ± 0.006**	**3.30 ± 2.34**
		32	**0.140 ± 0.075**	**0.014 ± 0.006**	**3.89 ± 2.97**
		128	**0.151 ± 0.082**	**0.016 ± 0.007**	**4.55 ± 3.65**
256	CFM	8	0.461 ± 0.007	0.049 ± 0.007	28.45 ± 4.63
		32	0.402 ± 0.049	0.047 ± 0.006	28.95 ± 10.71
		128	0.390 ± 0.069	0.045 ± 0.012	30.57 ± 13.24
	DFM	8	0.216 ± 0.049	0.029 ± 0.008	9.75 ± 5.07
		32	0.228 ± 0.045	0.033 ± 0.008	12.63 ± 4.81
		128	0.255 ± 0.044	0.039 ± 0.009	15.80 ± 4.96
	Count Bridge	8	**0.087 ± 0.045**	**0.0093 ± 0.0022**	**1.58 ± 1.28**
		32	**0.092 ± 0.044**	**0.0099 ± 0.0021**	**1.68 ± 1.30**
		128	**0.105 ± 0.039**	**0.012 ± 0.001**	**1.98 ± 1.26**
512	CFM	8	0.569 ± 0.055	0.051 ± 0.006	49.32 ± 7.46
		32	0.471 ± 0.046	0.049 ± 0.005	50.69 ± 11.35
		128	0.438 ± 0.034	0.048 ± 0.005	53.70 ± 14.23
	DFM	8	0.261 ± 0.069	0.042 ± 0.015	30.49 ± 21.88
		32	0.288 ± 0.099	0.050 ± 0.019	44.53 ± 29.76
		128	0.319 ± 0.112	0.058 ± 0.022	55.03 ± 35.35
	Count Bridge	8	**0.081 ± 0.028**	**0.010 ± 0.002**	**1.46 ± 0.73**
		32	**0.085 ± 0.026**	**0.010 ± 0.001**	**1.79 ± 0.89**
		128	**0.113 ± 0.029**	**0.016 ± 0.003**	**3.53 ± 2.27**

D.2 Architecture, Training, and Sampling

D.2.1 Inputs and tokenization

We model spot-level *counts* while conditioning on *image* context and diffusion *noise/time* tokens. Each training example provides

$$x_t^{\text{cnt}} \in \mathbb{Z}_{\geq 0}^{D_c}, \quad I_t \in \mathbb{R}^{C\times H\times W}, \quad t \in (0,1], \quad \varepsilon \in \mathbb{R}^{d_\varepsilon}, \quad y \in \{1,\dots,C_y\} \text{ (optional)}.$$

Table 7: Deconvolution Performance: W_2 vs Group Size and Dirichlet Concentration

Group Size (n)	$\alpha = 1$	$\alpha = 10$	$\alpha = 1000$
4	**0.0091 ± 0.0005**	0.010 ± 0.000	0.011 ± 0.000
8	0.011 ± 0.001	0.014 ± 0.001	0.016 ± 0.000
32	0.020 ± 0.002	0.023 ± 0.002	0.025 ± 0.002
128	0.050 ± 0.008	0.053 ± 0.006	0.057 ± 0.002

Table 8: Deconvolution Performance: EMD vs Group Size and Dirichlet Concentration

Group Size (n)	$\alpha = 1$	$\alpha = 10$	$\alpha = 1000$
4	**0.130 ± 0.006**	0.195 ± 0.025	0.207 ± 0.040
8	0.286 ± 0.023	0.363 ± 0.037	0.483 ± 0.010
32	0.530 ± 0.051	0.657 ± 0.095	0.921 ± 0.222
128	2.24 ± 0.58	2.22 ± 0.54	2.63 ± 0.14

Table 9: Deconvolution Performance: MMD vs Group Size and Dirichlet Concentration

Group Size (n)	$\alpha = 1$	$\alpha = 10$	$\alpha = 1000$
4	**0.011 ± 0.001**	0.011 ± 0.002	0.012 ± 0.004
8	0.016 ± 0.001	0.017 ± 0.002	0.021 ± 0.001
32	0.014 ± 0.003	0.020 ± 0.003	0.025 ± 0.010
128	0.037 ± 0.005	0.045 ± 0.007	0.044 ± 0.001

Images are patchified by a ViT-style embedder (`PatchEmbed`) into

$$X^{\text{img}} \in \mathbb{R}^{B\times N_{\text{img}}\times d}, \quad N_{\text{img}} = (H/P)\,(W/P),$$

while counts are converted into a small set of *count patches* using a learned projector (`CountPatchEmbedding`):

$$X^{\text{cnt}} = \text{reshape}\Big(\text{MLP}(x_t^{\text{cnt}}),\, [B, N_{\text{cnt}}, d]\Big) \;+\; E_{\text{cnt}},$$

with N_{cnt} learned "pseudo-patches" and E_{cnt} learnable positional embeddings.

We form auxiliary tokens for time, noise, and (optionally) class:

$$\underbrace{\tau}_{\text{time}} = \text{TimeMLP}\big(\text{timestep\_emb}(t)\big) \in \mathbb{R}^{B\times 1\times d}, \quad \underbrace{\eta}_{\text{noise}} = \text{NoiseMLP}\big(W_\varepsilon \varepsilon\big) \in \mathbb{R}^{B\times 1\times d},$$

and, if labels are used, $\ell = \text{Emb}(y) \in \mathbb{R}^{B\times 1\times d}$. Concatenating all tokens,

$$X^{(0)} = \big[\,\ell\,;\,\eta\,;\,\tau\,;\,X^{\text{img}}\,;\,X^{\text{cnt}}\,\big] \;+\; E_{\text{pos}} \in \mathbb{R}^{B\times(N_{\text{img}}+N_{\text{cnt}}+\text{extras})\times d},$$

with a single learned positional table E_{pos} covering all tokens.

D.2.2 U-ViT BACKBONE (FUSION AND DECODING)

We process $X^{(0)}$ with a U-Net–style ViT:

$$\underbrace{X^{(1)}, \ldots, X^{(L/2)}}_{\text{encoder (save skips)}} \xrightarrow{\text{mid Block}} \underbrace{\tilde{X}^{(L/2)}, \ldots, \tilde{X}^{(L)}}_{\text{decoder (with skips)}},$$

where each `Block` is a standard MHA +MLP transformer block with LayerNorm, and decoder blocks attend over skip connections. A final LayerNorm yields $X^{\text{out}} \in \mathbb{R}^{B\times(N_{\text{img}}+N_{\text{cnt}}+\text{extras})\times d}$.

We then drop the auxiliary tokens and split modalities:

$$X^{\text{img}}_{\text{out}} = X^{\text{out}}[:, \text{ img range}, :], \qquad X^{\text{cnt}}_{\text{out}} = X^{\text{out}}[:, \text{ cnt range}, :].$$

Count decoder. Count patches are decoded back to a vector via a small MLP head with nonnegativity enforced by `Softplus`:

$$\hat{x}_0 \;=\; \mathrm{Softplus}\Big(\mathrm{MLP}\big(\mathrm{flatten}(X_{\mathrm{out}}^{\mathrm{cnt}})\big)\Big) \in \mathbb{R}^{B\times D_c}.$$

This parameterizes $q_\theta(\cdot \mid x_t^{\mathrm{cnt}}, I_t, t, \varepsilon, y)$ for the distributional loss and the reverse count-bridge step.

D.2.3 Training objective and usage

We train the model to predict the distribution of X_0 (counts) given multimodal context under the bridge (X_t):

$$\mathcal{L}(\theta) \;=\; -\mathbb{E}_{t,(X_0,X_t)}\Big[\, S_\rho\big(q_\theta(\cdot \mid X_t^{\mathrm{cnt}}, I_t, t, \varepsilon, y),\, X_0^{\mathrm{cnt}}\big)\,\Big],$$

using the energy score S_ρ with $\rho(x,x') = \|x-x'\|_2^\beta$ ($\beta \in (0,2)$) and the standard unbiased U-statistic estimator with m samples from q_θ. Time and noise tokens implement the distributional diffusion conditioning; label tokens (if present) enable class-conditional modeling. During reverse sampling we draw $\tilde{X}_0 \sim q_\theta(\cdot \mid x_t^{\mathrm{cnt}}, I_t, t, \varepsilon, y)$ and update $x_{t-\Delta}$ using the exact binomial–hypergeometric count-bridge kernel (Prop. 3.1).

D.2.4 Implementation specifics

- **Patchification.** `PatchEmbed` uses patch size P on I_t (channels C), producing $N_{\mathrm{img}} = (H/P)(W/P)$ tokens of width d. `CountPatchEmbedding` projects D_c-dimensional counts to N_{cnt} tokens of width d with learned positional embeddings.
- **Auxiliary tokens.** Time token: `timestep_embedding` followed by a linear or MLP projector (`time_dim` controls concatenated components); noise token: linear to d then a 2-layer SiLU MLP; label token: lookup embedding if used. All tokens share a single learned positional table.
- **Backbone.** Depth L with $L/2$ encoder and $L/2$ decoder blocks; each block uses d-dimensional embeddings, h heads, MLP ratio r, LayerNorm, and (optionally) gradient checkpointing. Decoder blocks accept the matching encoder skip.
- **Heads.** Count head: 2-layer GELU MLP over the concatenated count tokens, ending with `Softplus`. Image head exists but is ignored for the loss.
- **No weight decay.** We exclude token positional tables and count-positional embeddings from weight decay, following ViT practice.

D.2.5 Optimization and hyperparameters

We use Adam, learning rate $\{2\times10^{-4}$, cosine warmup for 100 steps, EMA with 0.999, batch size 128, gradient clipping at 1.0. See configs for exact architecture specification.

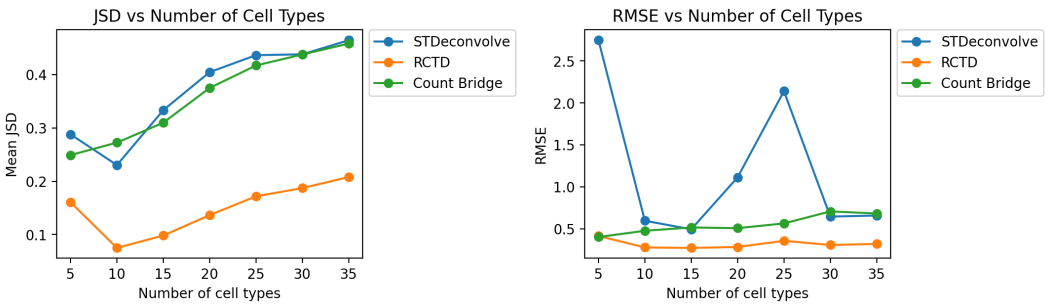
D.2.6 Sampling

At test time we follow Alg. 3 using the aggregates to ensure our sampled $\hat{x}_0$ exactly match the target sum at each intermediate time. We then apply the exact binomial–hypergeometric reverse kernel (Prop. 3.1) to obtain $x_{t_{k-1}}$.

D.3 Additional results

Comparison with reference-based methods Count Bridges and STDeconvolve are both reference-free methods: that is, they require only aggregate level data, and do not need a reference dataset of unit-level measurements. Many deconvolution methods, including RCTD (Cable et al., 2022) require a single-cell (non-spatial) reference dataset. These methods benefit from unit-level observations, and as such solve a more constrained problem – but require data which are not available in many settings.

Using the MERFISH benchmarking setup we also evaluate RCTD, and tabulate the results in Tab. 10. For evaluation, we use Jensen shannon Divergence (JSD) and RMSE metrics as described in (Li et al., 2023a). We find that Count Bridges perform similarly to RCTD (with a higher JSD but lower RMSE), despite the fact that Count Bridges do not have access to a reference dataset.



(a) (b)

Figure 5: Performance of RCTD, STDeconvolve and Count Bridge on MERFISH deconvolution across number of cell types using (a) Jensen Shannon Divergence (JSD) and (B) RMSE

Method	RMSE	JSD
STDeconvolve	2.746	0.289
RCTD	0.417	**0.161**
Count Bridge	**0.404**	0.250

Table 10: Cell-type error

In Fig. 5, we show the performance of spatial deconvolution methods across varying numbers of cell types. These results evaluate only the recovery of cell type proportions, and do not evaluate full count profiles. Note that RCTD has access to the single-cell level reference data, while STDeconvolve and Count Bridges are fit entirely using aggregate-level data and do not have access to single cell counts.

E Nucleotide-level gene expression modelling

E.1 Dataset

Nucleotide-level data preprocessing We use the Onek1k peripheral blood mononuclear cells (PBMC) 10X 3' scRNA-seq dataset, originally collected by Yazar et al. (2022). For our analysis, as we are interested in nucleotide-level counts rather than the gene-level counts provided with the initial publication, we use the preprocessed reads made available by Hingerl et al. (2024). The reads are aligned to the hg38 human reference genome. The resulting BAM files are filtered to include only high quality, UMI-deduplicated reads. The cell type annotations were used as provided in the original dataset.

Gene-level data preprocessing We construct the gene-level count matrices directly from our single-cell coverage matrices rather than following the typical single-cell gene expression preprocessing pipeline. In particular, for each annotated gene and each cell, we take the max count over the nucleotide-level coverage matrix as the count for the gene.

E.2 Architecture, Training, and Inference

E.2.1 Inputs and embeddings

We model nucleotide–level counts on a fixed window of length L=896. For each example we form

$$x_t \in \mathbb{Z}_{\geq 0}^{L}, \quad t \in (0,1], \quad z \sim \mathcal{N}(0, I_{d_z}), \quad c \in \{1, \ldots, C\}, \quad \text{seq} \in \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}, \texttt{N}\}^{L}.$$

Sequence context is embedded with a frozen Enformer encoder (EleutherAI checkpoint), yielding per–position embeddings

$$E(\text{seq}) \in \mathbb{R}^{L \times d_E}, \qquad d_E = 3072.$$

We tile the scalars across positions and concatenate

$$H^{(0)} \;=\; \big[\, E(\text{seq}) \,\|\, x_t \,\|\, t \,\|\, z \,\|\, \text{emb}(c) \,\big] \;\in\; \mathbb{R}^{L \times (d_E + 1 + 1 + d_z + d_c)},$$

with d_z=100 and d_c=14. A two–layer SELU MLP projects to the model width d:

$$X^{(0)} \;=\; \phi\big(W_2\, \phi(W_1 H^{(0)})\big) \in \mathbb{R}^{L \times d}.$$

E.2.2 LOCAL ATTENTION BACKBONE

We apply N_{attn} residual self–attention blocks (PyTorch `MultiheadAttention`, batch–first) with LayerNorm:

$$\tilde{X}^{(\ell)} \;=\; \mathrm{MHA}\big(X^{(\ell)}, X^{(\ell)}, X^{(\ell)}\big), \qquad X^{(\ell+1)} \;=\; \mathrm{LN}\big(\tilde{X}^{(\ell)} + X^{(0)}\big),$$

where the residual skip uses the pre–block $X^{(0)}$ as in the implementation.[3] We use N_{attn}=2 layers, d=`hidden_dim`, and h=4 heads. A linear projection followed by `softplus` produces a nonnegative per–position prediction

$$\hat{x}_0 \;=\; \mathrm{softplus}\big(W_{\text{out}} X^{(N_{\text{attn}})}\big) \in \mathbb{R}^{L}_{\geq 0}.$$

This parameterizes the conditional law $q_\theta(\,\cdot \mid x_t, t, z, c, \text{seq})$ used inside the count–bridge reverse kernel (Sec. 3).

E.2.3 LEARNED PROJECTION MODULE Π_ψ

When an aggregate constraint $a_0 = \sum_{i=1}^{L} x_{0,i}$ is observed, we refine $\hat{x}_0$ with a lightweight attention projector that operates across *positions*. We form

$$Y^{(0)} \;=\; \big[\, \hat{x}_0 \,\|\, x_t \,\|\, a_0 \,\|\, X^{(N_{\text{attn}})} \,\big] \in \mathbb{R}^{L \times (1+1+1+d)}.$$

A two–layer SELU MLP lifts to width d, then N_{proj}=2 self–attention layers (sequence–first API) with residual+LayerNorm are applied:

$$\tilde{Y}^{(m)} = \mathrm{MHA}(Y^{(m)}, Y^{(m)}, Y^{(m)}), \quad Y^{(m+1)} = \mathrm{LN}\big(\tilde{Y}^{(m)} + Y^{(0)}\big).$$

A linear head produces an additive correction which we re–softplus for nonnegativity:

$$\tilde{x}_0 \;=\; \mathrm{softplus}\big(W_{\text{proj}} Y^{(N_{\text{proj}})}\big) \;+\; \hat{x}_0.$$

At inference, when a_0 is present we use $\tilde{x}_0$ as the endpoint in the reverse step; otherwise we use $\hat{x}_0$.

E.2.4 TRAINING OBJECTIVES AND SCHEDULES

Distributional loss (energy score). We train q_θ with the energy score S_ρ on the conditional $X_0 \mid X_t{=}x_t$ (Sec. 3.2). For each example we draw m i.i.d. samples $\hat{x}_0^{(j)} \sim q_\theta(\cdot \mid x_t, t, z, c, \text{seq})$ via ancestral decoding of the per–position parameterization and estimate the U–statistic version of S_ρ with $\rho(x, x') {=} \|x{-}x'\|_2^\beta$ ($\beta \in (0, 2)$). We used m=2 in practice.

Aggregate–aware training. With probability p_{agg}=0.1 we attach an aggregate a_0 and route the forward pass through Π_ψ to obtain $\tilde{x}_0$, then compute the same energy score. This jointly trains Π_ψ to approximate sampling from the mean–conditional $X_0 \mid A(X_0){=}a_0, X_t, t$ while preserving the exact reverse transition of the count bridge.

Cell–type masking. To support both conditional and unconditional generation, we randomly mask the cell–type embedding with probability p_{mask} (set to zero vector). We used p_{mask}=0.1.

E.2.5 OPTIMIZATION AND HYPERPARAMETERS

We use Adam, learning rate $\{2{\times}10^{-4}$, cosine warmup for 100 steps, EMA with 0.999, batch size 128, gradient clipping at 1.0. See configs for exact architecture specification.

E.2.6 SAMPLING

At test time we follow Alg. 2: starting from x_1 we iterate $t_k \downarrow 0$. At each step we sample $\tilde{X}_0 \sim q_\theta(\cdot \mid x_{t_k}, t_k, z, c, \text{seq})$; if an aggregate is provided we replace with $\tilde{x}_0{=}\Pi_\psi(\hat{x}_0, a_0, x_{t_k})$. We then apply the exact binomial–hypergeometric reverse kernel (Prop. 3.1) to obtain $x_{t_{k-1}}$. This guarantees that trajectories remain within the discrete support while leveraging the learned distributional posterior. We use three function evaluations for all results in this application.

[3] Code uses a "global" residual $X \leftarrow X + X^{(0)}$ within each block. We retained this because it stabilized training with L=896.

E.3 Additional results

In Tab. E.3 we provide results for gene expression prediction performance, broken down by cell type.

Cell type	Baseline MSE	CB MSE
CD4 ET	3.596	**1.402**
NK	0.415	**0.364**
CD4 NC	3.382	**1.304**
CD8 S100B	2.619	**1.002**
CD8 ET	1.065	**0.540**
B IN	2.556	**1.091**
CD8 NC	3.381	**1.311**
B Mem	6.742	**3.416**
NK R	1.624	**0.781**
Mono NC	1.485	**0.752**
Mono C	1.253	**0.676**
DC	9.302	**4.475**
Plasma	10763.906	**10696.934**
CD4 SOX4	3.428	**1.323**