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 molecules detected. These measurements are often not at the desired resolution: while the unit of interest is typically a single cell, many measurement 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 addresss 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., 2023), 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

Diffusion models specify a time–indexed family of *bridge kernels* connecting $X_0 \sim p_0$ to a simple source distribution $X_1 \sim p_1$ (often Gaussian). There are two layers of structure: (i) an *unconditional forward process* $(X_t)_{t\in[0,1]}$ with kernels $K_{t|0}(x_t \mid x_0) = \mathrm{Law}(X_t \mid X_0 = x_0)$; (ii) for any $0 \le s \le t \le 1$, a family of *bridge kernels* $K_{s|0,t}(x_s \mid x_0, x_t) = \mathrm{Law}(X_s \mid X_0 = x_0, X_t = x_t)$.

Diffusion models require two consistency properties. First we require a bridge consistency identity.

$$\text{For any } 0 \le s \le t \le u \le 1, \quad K_{s|u}(x_s \mid x_u) = \int K_{s|t}(x_s \mid x_t)\, K_{t|u}(x_t \mid x_u)\, dx_t. \tag{1}$$

Thus multi-step sampling along any grid $u \to t \to s$ matches the single-step $u \to s$ bridge.

Second, the kernel must have a projective posterior:

$$K_{s|t}(x_s \mid x_t) = \int q_{0|t}(x_0 \mid x_t)\, K_{s|0,t}(x_s \mid x_0, x_t)\, dx_0, \tag{2}$$

where $q_{0|t}(x_0 \mid x_t) = \mathrm{Law}(X_0 \mid X_t = x_t)$. This identity expresses $K_{s|t}$ as a mixture over the posterior of the p_0 data. It is essential for denoising: during sampling, each predicted X_t changes the posterior $q_{0|t}$, so the reverse kernels must be projective under this posterior update.

Together, equation 1 and equation 2 lets us define a general diffusion approach. First we train a denoiser q_θ that approximates the posterior, $\tilde{X}_0 \sim q_\theta(\cdot \mid x_t, t) \approx \mathsf{Law}(X_0 \mid X_t{=}x_t)$, using tuples (t, X_t, X_0) drawn from the "global" bridge: sample $x_0 \sim p_0$, $x_1 \sim p_1$, $t \sim \mathrm{Unif}[0,1]$ and then $X_t \sim K_{t|0,1}(\cdot \mid x_0, x_1)$.

For sampling, pick a grid $1 = t_K > \cdots > t_0 = 0$, draw $X_1 \sim p_1$, set $X_{t_K} \leftarrow X_1$, sampling

$$\tilde{X}_0^{(k+1)} \sim q_\theta(\cdot \mid X_{t_{k+1}}, t_{k+1}), \qquad X_{t_k} \sim K_{t_k|0,t_{k+1}}(\cdot \mid \tilde{X}_0^{(k+1)}, X_{t_{k+1}}). \tag{3}$$

By our consistency properties, this multi–step procedure is equivalent to sampling directly from the $(0,1)$ bridge, so the model cannot drift out of the training distribution.

2.1 Diffusion as a bridge between noise and data

Let us consider the unconditional $K_{t|0}$ process $(X_t)_{t\in[0,1]}$ of the following form

$$X_t = \alpha(t)X_0 + B_t, \tag{4}$$

where $(B_t)_{t\in[0,1]}$ is a d-dimensional Gaussian process with non-decreasing standard deviation $\sigma(t)$, and $\alpha(t)$ a non-increasing function. Note that $\alpha(0){=}1$ and $\sigma(0){=}0$.

We want to define a process that interpolates smoothly between $X_0 \sim p_0$ 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 4. For $0 < s < t \le 1$, consider $(X_s)_{s\in[0,t]}$ conditioned on $X_t = x_t$ and $X_0 = x_0$. Then the conditional law $K_{s|0,t}(\cdot \mid x_0, x_t)$ is Gaussian and can be written*

$$X_s \overset{d}{=} \alpha(s)(1 - r(s,t))X_0 + \frac{\alpha(s)}{\alpha(t)} r(s,t) X_t + \sigma(s)(1 - r(s,t))^{1/2} Z, \tag{5}$$

where $Z \sim \mathcal{N}(0, \mathrm{Id})$ is independent of (X_0, X_t) and $r(s,t) = \frac{\alpha(t)^2\sigma(s)^2}{\alpha(s)^2\sigma(t)^2}$. In particular, the family $\{K_{s|0,t}\}_{0\le s\le t\le 1}$ defined by equation 5 satisfies equations 1 and 2.

Note that if $X_1 \sim \mathcal{N}(0, \mathrm{Id})$, $\alpha(1) = 0$ and $\sigma(1) = 1$ we have $X_t \overset{d}{=} \alpha(t)X_0 + \sigma(t)Z$. Furthermore, our equation 5 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 $\sigma(t)(1 - r(t))^{1/2} \to \gamma_t$.

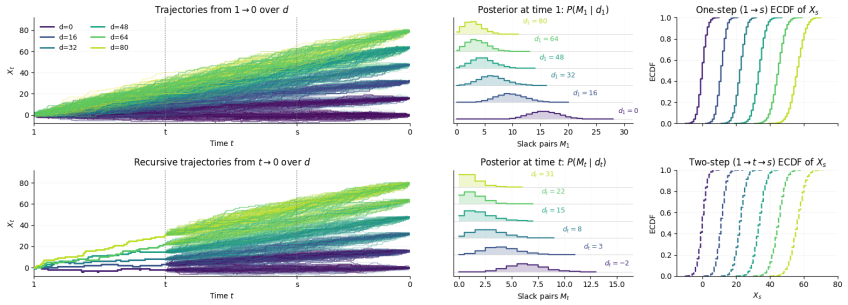


Figure 1: *Left:* Sample paths for several endpoint gaps d_1 (top). Fixing the prefix $[0, t]$ resample $(t, 1]$ by the recursive kernel (bottom). *Middle:* Bessel slack posteriors at initial and intermediate times. The slack M_t concentrates near 0 as $|d|$ grows. *Right:* ECDFs of X_s from a one–step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

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.

Infinitesimal. Consider a small backward step of size $\delta > 0$. The local bridge between times t and $t - \delta$ is Gaussian, so conditioned on $X_t = x$ we can write to first order in δ

$$X_{t-\delta} \mid X_t = x \;\approx\; x - \delta\, b(x,t) + \sqrt{\delta}\,\xi_t, \qquad \xi_t \sim \mathcal{N}\big(0, \Sigma(x,t)\big),$$

where b is the reverse-time drift and Σ is the diffusion covariance of the bridge.

The conditional law $X_{t-\delta} \mid X_t$ is Gaussian and can be computed in closed form:

$$b(x,t) \;=\; B_1(t)\, x + B_2(t)\, \mathbb{E}[X_0 \mid X_t = x] + b_0(t), \qquad \Sigma(x,t) \;=\; \Sigma_0(t).$$

The diffusion covariance depends only on t (from the Brownian increment), and the drift depends on the posterior $\mathsf{Law}(X_0 \mid X_t)$ only through its mean. This justifies learning the mean $q_\theta(x,t) \approx \mathbb{E}[X_0 \mid X_t = x]$ (equivalently, a score or velocity) as in standard diffusion models (Song et al., 2020).

Distributional. Following De Bortoli et al. (2025) we can learn the conditional law $q_\theta(\,\cdot \mid x_t, t) \approx \mathsf{Law}(X_0 \mid X_t{=}x_t)$, using any distribution learning approach. We can then sample and directly plug into the bridge

$$\tilde{X}_0^{(k+1)} \sim q_\theta(\,\cdot \mid X_{t_{k+1}}, t_{k+1}), \quad X_{t_k} \sim K_{t_k|0,t_{k+1}}\big(\,\cdot \mid \tilde{X}_0^{(k+1)}, X_{t_{k+1}}\big).$$

to sample the posterior. 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). 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).

3 Count Bridges

3.1 An integer bridge between distributions

Mirroring Sec. 2, we seek a bridge for integer-valued data. Instead of a Gaussian process, we use a pair of independent Poisson birth/death processes $(B_t)_{t\in[0,1]}$ and $(D_t)_{t\in[0,1]}$ that increment/decrement the counts. We define an increasing "jump-intensity" function $w : [0,1] \to \mathbb{R}_{\geq 0}$ with $w(0) = 0$, $w(1) = 1$, and then write the cumulative birth/death intensities as $\Lambda_\pm(t) = \lambda_\pm\, w(t)$ for some $\lambda_\pm > 0$ so $B_t \sim \mathrm{Poi}(\Lambda_+(t))$ and $D_t \sim \mathrm{Poi}(\Lambda_-(t))$. From here we can define the unconditional kernel $K_{t|0}$:

$$X_t = X_0 + B_t - D_t. \tag{6}$$

Denoting the displacement $d_t = X_t - X_0$, 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 determine the third:

$$N_t = |d_t| + 2M_t, \qquad B_t = \tfrac{1}{2}(N_t + d_t), \qquad D_t = N_t - B_t. \tag{7}$$

From the (N_t, B_t) perspective, Poisson superposition and thinning imply that, conditional on (N_t, B_t) at time t, the earlier counts (N_s, B_s) for $s < t$ can be sampled by a Binomial draw for N_s and a Hypergeometric draw for B_s. Switching to (M_t, d_t), a Poisson change of variables yields the slack posterior $M_t \mid d_t$, whose pmf has Bessel form (see Prop. A.5 in App. A). These two ingredients together give a count analogue of Proposition 2.1; the full derivation is in App. A.

Proposition 3.1. *Let $(X_t)_{t\in[0,1]}$ be given by equation 6. Now, consider $(X_s)_{s\in[0,t]}$ conditioned by $X_t = x_t$ and $X_0 = x_0$. Then, we have the Poisson Birth-Death bridge $K_{s|0,t}$:*

$$X_s \overset{d}{=} X_0 + B_s - D_s, \tag{8}$$

where we condition on $d_t = X_t - X_0$ and sample $M_t \mid d_t \sim$ Bes$(|d_t|; \Lambda_+(t), \Lambda_-(t))$, changing variables to N_t and B_t to sample B_s, and D_s which we can plug into equation 8:

$$N_s \mid N_t \sim \mathrm{Bin}\left(N_t, \frac{w(s)}{w(t)}\right), \quad B_s \mid (N_t, N_s, B_t) \sim \mathrm{Hyp}(N_t, B_t, N_s), \quad D_s = N_s - B_s. \tag{9}$$

The family $\{K_{s|0,t}\}_{0\le s\le t\le 1}$ defined by equation 8 satisfies equations 1 and 2.

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 the fast Bessel sampler of Devroye (2002) to enable sampling at scale.

In Fig. 1 we also see that as d_t grows the slack M_t concentrates near zero, so there is no slack. This means that Count Bridges are an instance of the static Schrödinger bridge problem (Léonard, 2013): they solve an entropy-regularized optimal transport. Let $\kappa = \sqrt{\lambda_+\lambda_-}$ be the jump intensity and $p^{\kappa}_{\mathrm{ref}}(x_0, x_1) = p_0(x_0)K^{\kappa}_{1|0}(x_1|x_0)$ be the joint law of (X_0, X_1) induced by the kernel. Over the space of couplings $\mathcal{C}(p_0, p_1) = \{C \text{ on } \mathcal{X}\times\mathcal{X} : C(\cdot, \mathcal{X}) = p_0,\ C(\mathcal{X}, \cdot) = p_1\}$, Count Bridges solve

$$C_\kappa \in \arg\min_{C\in\mathcal{C}(p_0,p_1)} \mathrm{KL}\big(C \,\|\, p^{\kappa}_{\mathrm{ref}}\big).$$

Letting $\kappa \to \infty$ yields the independent coupling $p_0 \otimes p_1$, but as $\kappa \downarrow 0$ we obtain

$$\mathrm{KL}\big(C \,\|\, p^{\kappa}_{\mathrm{ref}}\big) \;\approx\; \log\Big(\tfrac{2}{\kappa}\Big)\, \mathbb{E}_C|X_1 - X_0| - H(C),$$

so $\kappa \to 0$ recovers discrete OT with cost $|x_1 - x_0|$ (see App. A.2).

This echoes the Gaussian case (Sec. 2) where we define $\sigma = \sigma(1)$ and $p^{\sigma}_{\mathrm{ref}}$, and as $\sigma \downarrow 0$

$$\mathrm{KL}\big(C \,\|\, p^{\sigma}_{\mathrm{ref}}\big) \;\approx\; \tfrac{1}{2\sigma^2}\, \mathbb{E}_C\|X_1 - X_0\|^2 - H(C),$$

so $\sigma \to 0$ recovers quadratic OT, while $\sigma \to \infty$ again gives $p_0 \otimes p_1$ (Shi et al., 2023). Thus the bridge parameters κ (count) and σ (Gaussian) play the same role as entropy–regularization strengths.

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. We can use cross-entropy with Count Bridges (we test this, see App. D.1), but it has two issues: first, it does not incorporate the lattice structure; second, cross-entropy cannot model the joint of $X_s \mid X_t$ without exponential cost in dimension, so cross entropy is usually factorized, modeling each coordinate of $X_s \mid X_t$ independently or autoregressively. Specializing to count data we can go beyond cross-entropy by using a proper scoring rule that (i) incorporates the geometry and (ii) enables modeling of the joint.

Formally, let (X_0, X_t) denote a training pair from $K_{t|0,1}$ at time $t \in [0, 1]$, and let $q_\theta(\cdot \mid x_t, t)$ be our denoiser. We train q_θ using a strictly proper distributional scoring rule (Gneiting & Raftery, 2007;

Require: dataset $(\mathbf{x}_0, \mathbf{x}_1)$, $w(\cdot), \Lambda_\pm(\cdot)$
for each minibatch **do**
  sample $(x_0, x_1) \sim (\mathbf{x}_0, \mathbf{x}_1)$
  $t \sim \mathrm{Unif}[0,1]$
  $d_1 \leftarrow x_1 - x_0$
  $M_1 \sim \mathrm{Bes}(|d_1|;\, \Lambda_+(1), \Lambda_-(1))$
  $N_1 \leftarrow |d_1| + 2M_1$
  $B_1 \leftarrow \frac{1}{2}(N_1 + d_1)$
  $N_t \sim \mathrm{Bin}\big(N_1,\, w(t)\big)$
  $B_t \sim \mathrm{Hyp}\big(N_1,\, B_1,\, N_t\big)$
  $x_t \leftarrow x_1 - 2(B_1 - B_t) + (N_1 - N_t)$
  update θ on $\mathcal{L}(\theta)$
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 $\hat{x}_0 \sim q_\theta(\cdot \mid x_{t_k}, t_k)$
  $d_{t_k} \leftarrow x_{t_k} - \hat{x}_0$
  $M_{t_k} \sim \mathrm{Bes}(|d_{t_k}|;\, \Lambda_+(t_k), \Lambda_-(t_k))$
  $N_{t_k} \leftarrow |d_{t_k}| + 2M_{t_k}$
  $B_{t_k} \leftarrow \frac{1}{2}(N_{t_k} + d_{t_k})$
  $r \leftarrow w(t_{k-1})/w(t_k)$
  $N_{t_{k-1}} \sim \mathrm{Bin}\big(N_{t_k},\, r\big)$
  $B_{t_{k-1}} \sim \mathrm{Hyp}\big(N_{t_k},\, B_{t_k},\, N_{t_{k-1}}\big)$
  $x_{t_{k-1}} \leftarrow x_{t_k} - 2(B_{t_k} - B_{t_{k-1}}) + (N_{t_k} - N_{t_{k-1}})$
end for
return x_{t_0}

Algorithm 2: Sampling Poisson–BD Bridge

De Bortoli et al., 2025). Fix a negative-type semimetric ρ on $\mathbb{Z}^D$ (all our experiments focus on the $\rho(x, x') = \|x - x'\|_2^\beta$ with $\beta = 1$). For any distribution p and outcome y, the energy score is

$$S_\rho(p, y) = \tfrac{1}{2}\,\mathbb{E}_{X,X'\sim p}\big[\rho(X, X')\big] - \mathbb{E}_{X\sim p}\big[\rho(X, y)\big] \text{ and } \mathcal{L}(\theta) = \mathbb{E}_{X_0, X_t, t}\Big[S_\rho\big(q_\theta(\cdot \mid X_t, t),\, X_0\big)\Big]$$

which is strictly proper when ρ is characteristic. Taking m i.i.d. samples $\hat{x}^{(j)} \sim q_\theta(\cdot \mid x_t, t)$ we can use the plugin estimator: $\widehat{S}_\rho = \frac{1}{m(m-1)} \sum_{j \neq j'} \frac{1}{2}\rho(\hat{x}^{(j)}, \hat{x}^{(j')}) - \frac{1}{m}\sum_{j=1}^m \rho\big(\hat{x}^{(j)}, x_0\big)$.

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, $a_0 = \sum_{g=1}^G x_{g0} \in \mathbb{Z}$, not the unit–level vector $\mathbf{x}_0$. Our goal is to learn a count bridge $q_\theta(x_0 \mid x_t, t, z)$ 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 $q_\theta(\cdot \mid x_t, t, z)$ defines an i.i.d. product prior over $\mathbf{X}_0 = (X_{10}, \ldots, X_{G0})$. Conditioning on the aggregate yields

$$Q_\theta(\mathbf{X}_0 \mid a_0, x_t, t, z) \;\propto\; \Big[\prod_{g=1}^G q_\theta(X_{g0} \mid x_t, t, z)\Big]\, \mathbf{1}\{A(\mathbf{X}_0) = a_0\}.$$

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.

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

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

We could then use the sampled $\mathbf{x}_0^\star$ as latent variables to sample x_t between $(\mathbf{x}_0^\star, \mathbf{x}_1)$ using the unit–level kernel $K_{t|0,1}$ from 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

[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
for $k = K, K-1, \ldots, 2$ **do**
  Sample $\hat{\mathbf{x}}_{0,t_k} \sim q_\theta(\cdot \mid \mathbf{x}_{t_k}, t_k, z)$
  $\tilde{\mathbf{x}}_{0,t_k} \leftarrow \Pi(\hat{\mathbf{x}}_{0,t_k}, a_0, z)$
  Update $\mathbf{x}_{t_{k-1}}$ by running the reverse step
    using steps 4–10 of Alg. 2, with $\tilde{\mathbf{x}}_{0,t_k}$
end for
$\mathbf{x}_0^{\approx} \leftarrow$ sample and project $\hat{\mathbf{x}}_{0,t_1}$
return $\mathbf{x}_0^{\approx}$

Algorithm 3: Guided Sampling to for $x_0^{\approx}$

Require: $(\mathbf{x}_1, a_0, z)$, $w(\cdot), \Lambda_{\pm}(\cdot)$, q_θ, Π
for each minibatch **do**
  E-step: Sample latent $\mathbf{x}_0^{\approx}$ from
    $\mathbf{x}_1$ conditional on a_0 via Alg. 3
  M-step: $t \sim \mathrm{Unif}[0, 1]$
    Sample $\mathbf{x}_t$ via the forward bridge on
      $(\mathbf{x}_0^{\approx}, \mathbf{x}_1)$ using steps 4–10 of Alg. 1
  Update θ using the gradient of $-\mathcal{L}_{\text{agg}}(\theta)$
end for

Algorithm 4: Training with Aggregate Supervision

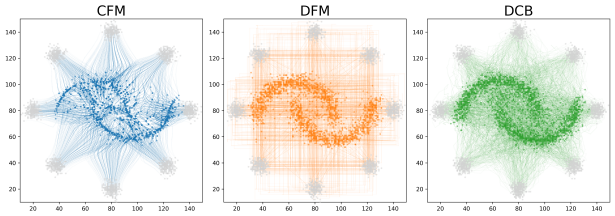


Figure 2: A scaled and rounded variant of the classic 8 gaussian to two moons task. Here we compare the trajectories of continuous flow matching, discrete flow matching, and count bridges. CB achieves the lowest W_2, MMD, and EMD, see Table 6.

$\tilde{\mathbf{x}}_0$, 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:

$$S_\rho^A(p, a) = \tfrac{1}{2}\,\mathbb{E}_p\big[\rho(A(\mathbf{X}), A(\mathbf{X}'))\big] - \mathbb{E}_p\big[\rho(A(\mathbf{X}), a)\big] \text{ and } \mathcal{L}_{\text{agg}}(\theta) = \mathbb{E}_{A_0, \mathbf{X}_t, t}\Big[S_\rho^A\big(q_\theta(\cdot \mid \mathbf{X}_t, t, z),\ A_0\big)\Big]$$

with the plug-in obtained by sampling $\hat{\mathbf{X}}_0^{(j)} \sim q_\theta(\cdot \mid \mathbf{X}_t, t, z)$ and forming $\hat{a}^{(j)} = A(\hat{\mathbf{X}}_0^{(j)})$.

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 $Q_\theta(\cdot \mid A(\mathbf{X}_0) = a_0)$. While this is intractable, we can derive a principled approximation.

Proposition 4.1 (First–order aggregate projection)**.** *Let $A(\mathbf{X}_0)$ be the aggregate, and let P_θ be the prior law w $\mathbf{X}_0$. Under the regularity conditions in App. B.1, the aggregate–conditional law $Q_\theta(\cdot \mid A_0 = a_0)$ admits a first–order exponential tilt. The corresponding KL projection*

$$\Pi(\mathbf{x}_0) = \arg \min_{\mathbf{y}_0:\, A(\mathbf{y}_0) = a_0} D_{\mathrm{KL}}(\mathbf{y}_0 \| \mathbf{x}_0)$$

gives a first–order approximation to $Q_\theta(\cdot \mid A_0 = a_0)$. For an elementwise sum $A(\mathbf{x}_0) = \sum_g x_{g0}$ this projection is the simple scaling $\Pi(x_0)_g = a_0 x_{g0} / (\sum_{g'} x_{g'0})$.

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.

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.

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., 2023). 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. DestVI (Lopez et al., 2022), which outputs count profiles but requires a reference, and STDeconvolve (Miller et al., 2022) which does not require a reference but outputs cluster-level predictions, both take steps toward this goal.

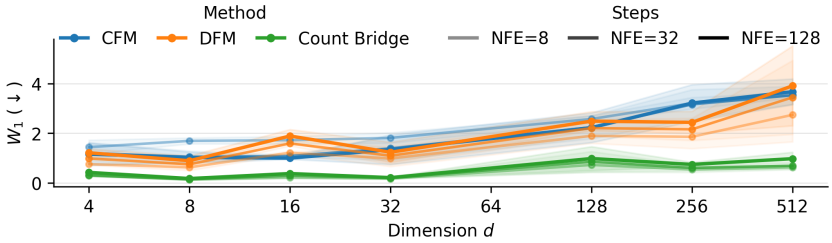


Figure 3: CFM, DFM, and CB on our low-rank mixture of Gaussians transport experiment across dimensions and NFE. See App. D.2 for full details.

6 Applications

We evaluate with three distributional metrics: the Energy score, the Wasserstein-2 distance, and the MMD (RBF). For deconvolution, we evaluate cell-type proportion predictions using RMSE, the Jensen-Shannon Divergence (JSD), and Spearman correlation following Li et al. (2023). Synthetic tasks have std. errors over 3 training seeds; main applications have std. errors 3 over inference seeds.

6.1 Synthetic Distributions

Here, 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. D.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. D.2 for more).

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 D.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).

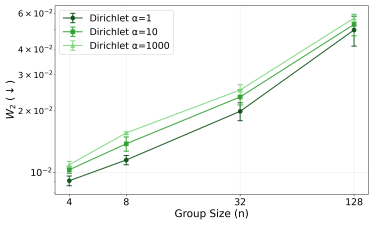


Figure 4: Deconvolution of the low-rank Gaussian mixture across different group sizes and levels of between-group heterogeneity.

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 relate sequence and expression, the most prominent of which, such as Enformer (Avsec

Method	Bulk MSE	CT MSE
Fine-tuned Enformer	2.590	3.142
Count Bridge	**0.601** ± 0.000	**1.410** ± 0.002

Table 1: Nucleotide-level MSE for bulk and bulked cell-type (CT) specific predictions.

Comparison	MMD	$\mathbf{W_2}$	Energy
Bulk mean	0.515	0.208	56.800
Count Bridge	**0.446** ± 0.000	**0.182** ± 0.001	**28.583** ± 0.003

Table 2: Gene expression count profile deconvolution error for bulk RNA sequencing data.

et al., 2021), are Transformer-based models that predict expression from sequence. More recent work has explored fine-tuning Enformer on single-cell data (Hingerl et al., 2024). On the other hand, there is a mature literature on deconvolving bulk RNA-seq (Newman et al., 2019; Wang et al., 2019). These methods operate at the gene (rather than nucleotide) level, leveraging bulk cell-type profiles or single-cell references to deconvolve bulk profiles into cell-type proportions (not count profiles).

We use CBs to jointly model sequence and single-cell expression counts in scRNA-seq data, and to enable nucleotide-level deconvolution of bulk profiles. To validate CBs in this setting, we demonstrate two key results. First, we show that CBs trained on single-cell data produce meaningful count profiles and outperform a fine-tuned Enformer model on sequence-to-expression prediction. Second, we show that conditioning CBs on bulk profiles enables deconvolution of bulk gene expression into inferred single-cell gene expression profiles. We validate these deconvolved profiles distributionally and show that they achieve state-of-the-art performance relative to cell-type proportion deconvolution models.

Method	JSD	RMSE	Spearman
CIBERSORTx	0.194	0.109	0.079
MuSiC	0.313	0.140	0.186
Count Bridge	**0.113** ± 0.001	**0.073** ± 0.000	**0.267** ± 0.005

Table 3: Cell-type proportion deconvolution error for nucleotide level bulk RNA sequencing data.

Modeling sequence and single-cell counts We train CBs on PBMC scRNA-seq counts at nucleotide resolution using 10^6 cells across 10^3 donors (Yazar et al., 2022). Each training example corresponds to a nucleotide position in a single cell, and is represented by the noisy count x_t and diffusion time t from the CB forward process, a cell-type embedding, a local genomic context z obtained by encoding the surrounding DNA sequence with Enformer, and i.i.d. noise ζ for the distributional loss. These features are concatenated and passed through residual multi–head attention blocks and a final softplus head that parameterizes the conditional count distribution $X_0|X_t, t, z$. The model is trained directly on unit-level (single-cell) expression profiles rather than only on aggregated counts. During training we randomly mask cell-type labels so that the model supports both unconditional and cell-type-conditional sampling at test time.

Learned projection for deconvolution Since we have unit-level data we can learn a better projection operator than the simple rescaling function in Prop. 4.1. We augment the CB with a small projection module Π_ψ, an attention block operating on each nucleotide (represented by z) across cells in the batch. Given an initial CB prediction $\hat{x}_0$, an observed aggregate a_0, and the noisy state x_t, the module outputs $\tilde{\mathbf{x}}_0 = \Pi_\psi(\hat{\mathbf{x}}_0, a_0, \mathbf{x}_t, z)$, we train this using the distributional loss to learn to sample $X_0 \mid A(X_0){=}a_0, X_t, t$. To support both unconditional and aggregate-conditioned inference, we apply the projection module only on a random 10% of training examples where a_0 is provided.

Bulk gene expression We first evaluate the ability of our model to predict expression from sequence, both unconditionally and conditional on cell type. As a baseline, we use an Enformer model 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).

Deconvolved profiles We can use this unit-level model for deconvolution tasks: we can condition on an aggregate (bulk profile) to sample single-cell profiles from the model while matching that aggregate. We next evaluate the ability of CBs to deconvolve mixtures of cell types from held-out individuals. We held out 10% of patients from our training set and synthetically bulked these patients. Since we have the ground truth data, we can then evaluate deconvolution quality. We first evaluate the distributional quality of these predictions against the bulk mean, further validating the CB count profiles (Table 2). As a more robust set of baselines, we compare to CIBERSORTx (Newman et al., 2019) and MuSiC (Wang et al., 2019). To facilitate comparison, we aggregate our nucleotide-level predictions into gene counts and assign each of our deconvolved cells to the closest

cell type. CBs achieve better performance on JSD, RMSE, and Spearman correlation while providing nucleotide-level counts (Table 3). In App. E we plot the UMAP for qualitative comparison.

6.3 Deconvolving spatial transcriptomic spots into single-cell counts

Next, we show how CBs can be used to infer single cell gene expression profiles from spot-level aggregates in spatial transcriptomic data. In spatial transcriptomic data generated by 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). CBs provide a natural way to leverage this cell-level side information to deconvolve aggregate count data.

Modeling spatial aggregates We train CBs on a MERFISH mouse brain dataset (Vizgen, 2021), which is resolved at the single-cell level, and artificially 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. Following the notation in Sec. 4, the spot-level counts can be treated as aggregates a_0, and single-cell images can be treated as unit-level side information z. In this application, we never observe single-cell count profiles, only spot-level aggregates and the single-cell images. We leverage a UViT (Bao et al., 2023) extended to incorporate count and noise patches (see App. F). We use a simple source distribution $X_1 \sim \mathrm{Poi}(10)$.

Cell type proportions We benchmark CBs against 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. (2023) (see Appendix F 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 by comparing the predicted cell type proportions to the true cell type proportions per spot. For CBs, which provide single-cell count profile predictions rather than cell type proportions, we assign each predicted count profile its nearest neighbor cell type in order to compare against STDeconvolve. CBs outperforms STDeconvolve on both the JSD and the RMSE (Table 4).

Method	JSD	RMSE	Spearman
STDeconvolve	0.288	0.177	0.255
Count Bridge	**0.231** ± 0.002	**0.110** ± 0.001	**0.332** ± 0.001

Table 4: Cell-type proportion deconvolution error for spatial transcriptomics.

Count profiles We next evaluate the quality of the count profiles inferred by CBs. 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, CBs outperform the spot-level mean baseline (see Table 5), showing CBs can learn meaningful unit-level distributions from real-world aggregate data. In App. F we provide a more detailed biological evaluation of the cell types and pathways in our generated data, alongside the UMAP to facilitate qualitative comparison.

Comparison	MMD	$\mathbf{W_2}$	Energy
Spot mean	0.409	0.030	41.717
Count Bridge	**0.203** ± 0.000	**0.017** ± 0.000	**8.903** ± 0.014

Table 5: Gene expression count profile deconvolution error for spatial transcriptomics

7 Conclusion

Count Bridges offer a tractable, discrete-native alternative to continuous diffusion models, unifying direct count generation with deconvolution from aggregates. We 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. 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.

C Domínguez Conde, C Xu, L B Jarvis, D B Rainbow, S B Wells, T Gomes, S K Howlett, O Suchanek, K Polanski, H W King, L Mamanova, N Huang, P A Szabo, L Richardson, L Bolt, E S Fasouli, K T Mahbubani, M Prete, L Tuck, N Richoz, Z K Tuong, L Campos, H S Mousa, E J Needham, S Pritchard, T Li, R Elmentaite, J Park, E Rahmani, D Chen, D K Menon, O A Bayraktar, L K James, K B Meyer, N Yosef, M R Clatworthy, P A Sims, D L Farber, K Saeb-Parsy, J L Jones, and S A Teichmann. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science (New York, N.Y.)*, 376(6594):eabl5197, 13 May 2022. ISSN 0036-8075,1095-9203. doi: 10.1126/science.abl5197.

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(1):1548, 21 March 2023. 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.

Romain Lopez, Baoguo Li, Hadas Keren-Shaul, Pierre Boyeau, Merav Kedmi, David Pilzer, Adam Jelinski, Ido Yofe, Eyal David, Allon Wagner, Can Ergen, Yoseph Addadi, Ofra Golani, Franca Ronchese, Michael I Jordan, Ido Amit, and Nir Yosef. DestVI identifies continuums of cell types in spatial transcriptomics data. *Nature biotechnology*, 40(9):1360–1369, September 2022. ISSN 1087-0156,1546-1696. doi: 10.1038/s41587-022-01272-8.

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.

Xuran Wang, Jihwan Park, Katalin Susztak, Nancy R Zhang, and Mingyao Li. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nature communications*, 10(1): 380, 2019.

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

We now formalize where the Binomial and Hypergeometric terms in equation 9 come from. We work directly with the birth–death representation introduced above.

Let $\lambda_+, \lambda_- > 0$ and let $w : [0,1] \to \mathbb{R}_{\geq 0}$ be an increasing "jump–intensity" shape function with $w(0) = 0$ and $w(1) = 1$. Define the cumulative birth/death intensities

$$\Lambda_+(t) = \lambda_+ \, w(t), \qquad \Lambda_-(t) = \lambda_- \, w(t),$$

and let $(B_t)_{t\in[0,1]}$ and $(D_t)_{t\in[0,1]}$ be independent (non-homogeneous) Poisson processes with these cumulants. Set

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

so that the unconditional forward kernel $K_{t|0}$ is the same as in equation 6.

Write the total jump count as

$$N_t = B_t + D_t, \qquad t \in [0,1].$$

Standard properties of Poisson superposition imply that (N_t) is a (non-homogeneous) Poisson process with cumulative intensity

$$\Lambda(t) = \Lambda_+(t) + \Lambda_-(t) = (\lambda_+ + \lambda_-)\, w(t).$$

Conditional on $N_1 = n$, the n unordered jump times are i.i.d. with cdf

$$\mathbb{P}(T \le t \mid N_1 \ge 1) \;=\; w(t), \qquad t \in [0,1],$$

so w is precisely the time–rescaling that makes the jump times uniform on $[0,1]$.

We now condition on endpoints $X_0 = x_0$, $X_1 = x_1$ and define

$$d_1 = x_1 - x_0, \qquad N_1 = B_1 + D_1, \qquad B_1 = \tfrac{1}{2}(N_1 + d_1), \qquad D_1 = N_1 - B_1.$$

Given (N_1, B_1), we can view the N_1 jumps as a set of N_1 points on $[0,1]$, each labelled $+1$ (birth) or -1 (death). The superposition and thinning properties give two key facts:

We now record the Binomial and Hypergeometric structure of the jump counts.

Lemma A.1 (Binomial–Hypergeometric structure of the birth–death bridge)**.** *Fix $t \in (0,1]$. Conditional on $N_1 = n$ and $B_1 = b$, let $0 < T_{(1)} < \cdots < T_{(n)} \le 1$ denote the ordered jump times and let $(L_1, \ldots, L_n) \in \{+1, -1\}^n$ be the corresponding jump labels, with $\sum_{k=1}^n \mathbf{1}\{L_k = +1\} = b$. Then:*

1. Binomial structure. *The unordered jump times are i.i.d. on $(0,1]$ with cdf w, so*

$$N_t \mid N_1 = n \;\sim\; \mathrm{Bin}\big(n,\, w(t)\big).$$

2. Hypergeometric structure. *Conditional on $(N_1, B_1, N_t) = (n, b, m)$, the labels $(L_1, \ldots, L_n)$ are exchangeable given $B_1 = b$, and the m jumps with times in $(0,t]$ correspond to a uniformly random m–subset of $\{1, \ldots, n\}$. Hence*

$$B_t \mid (N_1, B_1, N_t) = (n, b, m) \;\sim\; \mathrm{Hyp}\big(n,\, b,\, m\big),$$

and therefore $D_t = N_t - B_t$.

More generally, for $0 < s < t < 1$ define $r = w(s)/w(t)$. Conditional on $(N_t, B_t) = (n, b)$, the n jumps in $(0,t]$ form a non-homogeneous Poisson process with cumulative intensity $\Lambda(\cdot)/\Lambda(t)$, so the N_s jumps in $(0,s]$ satisfy

$$N_s \mid N_t = n \;\sim\; \mathrm{Bin}\big(n,\, r\big),$$

and, by the same exchangeability argument applied within $(0,t]$,

$$B_s \mid (N_t, B_t, N_s) = (n, b, m) \;\sim\; \mathrm{Hyp}\big(n,\, b,\, m\big), \qquad D_s = N_s - B_s.$$

A.1.2 Composition conditional on the slack

We now show that the Binomial–Hypergeometric construction is closed under composition when we condition on a fixed slack value.

Conditional on (X_0, X_1, M_1) (equivalently on (N_1, B_1)), the total number of jumps N_t and the number of births B_t at any intermediate time t are governed by the Binomial–Hypergeometric structure in Lemma A.1. We now record two elementary composition facts which we will use with

$$p = w(t), \qquad q = \frac{w(s)}{w(t)}, \qquad 0 < s < t < 1.$$

Lemma A.2 (Binomial composition)**.** *Let $K \sim \mathrm{Bin}(N, p)$ and, conditional on K, let $L \sim \mathrm{Bin}(K, q)$. Then $L \sim \mathrm{Bin}(N, pq)$. Moreover, (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. We begin by writing the joint distribution directly from the model:

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

Apply the standard combinatorial identity

$$\binom{N}{k}\binom{k}{\ell} = \binom{N}{\ell}\binom{N-\ell}{k-\ell},$$

to obtain the factorization

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

To get the marginal law of L, sum the joint pmf over all $k \geq \ell$. Let $m = k - \ell$; then m ranges from 0 to $N - \ell$, and

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

All terms not depending on m factor out:

$$\mathbb{P}(L=\ell) = \binom{N}{\ell}(pq)^\ell (1-pq)^{N-\ell} \sum_{m=0}^{N-\ell} \binom{N-\ell}{m} a^m b^{(N-\ell)-m},$$

where

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

Since $a + b = 1$, the inner sum is exactly the Binomial theorem,

$$\sum_{m=0}^{N-\ell} \binom{N-\ell}{m} a^m b^{(N-\ell)-m} = (a+b)^{N-\ell} = 1.$$

Thus

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

which is the pmf of $\mathrm{Bin}(N, pq)$. □

Lemma A.3 (Hypergeometric composition)**.** *Fix an urn with N balls of which B are marked "success". Draw $N_t = k$ balls without replacement and record $B_t = b$ successes. From the k drawn balls, draw $N_s = j \leq k$ without replacement and record $B_s = a$ successes. Then the marginal law of (N_s, B_s) 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 is the pmf of the single hypergeometric draw $\mathrm{Hyp}(N, B, j)$.

Proof. Fix the total population of N items with B successes. The first draw selects a subset $S_t \subset \{1, \ldots, N\}$ of size $k = N_t$ uniformly among all $\binom{N}{k}$ such subsets. From S_t the second draw selects a subset $S_s \subset S_t$ of size $j = N_s$ uniformly among all $\binom{k}{j}$ subsets of S_t.

Every subset S_s of size j can be formed in at least one two–step sequence of the form $S_s \subset S_t \subset \{1, \ldots, N\}$. Moreover, under simple sampling without replacement, all such two–step sequences occur with equal probability:

$$\mathbb{P}(S_s = A) = \sum_{\substack{T \supset A \\ |T|=k}} \frac{1}{\binom{N}{k}} \cdot \frac{1}{\binom{k}{j}} = \frac{\binom{N-j}{k-j}}{\binom{N}{k}\binom{k}{j}} \qquad \text{for any size-}j \text{ subset } A.$$

The right-hand side does not depend on A, so conditional on $N_s = j$ every j–subset of $\{1, \dots, N\}$ is equally likely. There are $\binom{N}{j}$ such subsets.

Among these $\binom{N}{j}$ subsets, exactly

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

subsets contain a successes and $j - a$ failures. Hence

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

which is the pmf of the hypergeometric distribution $\mathrm{Hyp}(N, B, j)$. □

We can now state composition of the birth–death bridge at fixed slack.

Theorem A.4 (Bridge composition conditional on slack)**.** *Fix $0 < s < t < 1$ and endpoints $X_0 = x_0$, $X_1 = x_1$, and slack $M_1 = m$ (equivalently, fix (N_1, B_1) via equation 7). Let $w : [0, 1] \to [0, 1]$ be the time–rescaling function from the birth–death construction.*

Define the fixed–slack bridge kernels

$$K^{(m)}_{a|0,b}(x_a \mid x_0, x_b) \;=\; \mathbb{P}\big(X_a = x_a \,\big|\, X_0 = x_0, X_b = x_b, M_1 = m\big), \qquad 0 \le a \le b \le 1.$$

Then the two–stage rule $(1 \to t \to s)$ yields the same law for (N_s, B_s) as the single–stage rule $(1 \to s)$, and the fixed–slack kernels satisfy the bridge–consistency identity

$$K^{(m)}_{s|0,1}(x_s \mid x_0, x_1) = \sum_{x_t} K^{(m)}_{s|0,t}(x_s \mid x_0, x_t)\, K^{(m)}_{t|0,1}(x_t \mid x_0, x_1). \tag{10}$$

Proof. Working conditional on (N_1, B_1) (equivalently on $(x_0, x_1, M_1 = m)$), the Binomial–Hypergeometric structure of Lemma A.1 gives

$$N_t \mid N_1 \sim \mathrm{Bin}\big(N_1, w(t)\big), \qquad N_s \mid N_t \sim \mathrm{Bin}\big(N_t, w(s)/w(t)\big).$$

Applying Lemma A.2 with $p = w(t)$ and $q = w(s)/w(t)$ yields

$$N_s \mid N_1 \sim \mathrm{Bin}\big(N_1, w(s)\big),$$

which coincides with the direct $(1 \to s)$ rule.

For the colour counts, conditional on (N_1, B_1) and N_t, the N_t jumps in $(0, t]$ form a finite population with B_t births (successes) and $N_t - B_t$ deaths (failures). Within this population, the two–stage sampling $(t \to s)$ is exactly the nested hypergeometric scheme of Lemma A.3, so the marginal law of (N_s, B_s) coincides with that of a single hypergeometric draw $\mathrm{Hyp}(N_1, B_1, N_s)$ from the original N_1 jumps. This is the direct $(1 \to s)$ rule.

Thus, at fixed $(x_0, x_1, M_1 = m)$, the two–step transition $(1 \to t \to s)$ and the one–step transition $(1 \to s)$ agree in law for (N_s, B_s), and hence for $X_s = x_0 + B_s - D_s$. Equivalently, the conditional distributions $\mathbb{P}(X_s = x_s \mid X_0 = x_0, X_1 = x_1, M_1 = m)$ obey the kernel identity equation 10, which is exactly the fixed–slack bridge consistency for $K^{(m)}_{a|0,b}$. □

A.1.3 Integrating over the slack: the observable bridge

In the previous section we established that, *conditional on a fixed slack value*, the birth–death (BD) bridge satisfies exact composition: for fixed (X_0, X_t, M_t) the $(t \to s)$ transition coincides with the direct $(t \to s)$ rule defined by the Binomial–Hypergeometric structure. However, our generative model does not include a latent slack variable: at time t we observe only $X_t = X_0 + B_t - D_t$, not the pair (B_t, D_t) or their slack $M_t = \min(B_t, D_t)$.

To obtain an observable Markov bridge we therefore *resample the slack at each time* from its posterior distribution

$$\pi_t(m \mid d_t) = \mathbb{P}\big(M_t = m \mid d_t = X_t - X_0\big),$$

and then apply the conditional BH transition at that m. This subsection derives the explicit form of $\pi_t(m \mid d_t)$ and, crucially, proves its closure under the time-rescaling $w(t)$ of the BD reference. This closure ensures that the mixture over M_t interacts correctly with the BH transitions, enabling the observable process to satisfy the bridge consistency equation 1.

The Slack Posterior

Recall from the BD reference that

$$B_t \sim \mathrm{Poi}(\Lambda_+(t)), \qquad D_t \sim \mathrm{Poi}(\Lambda_-(t)), \qquad \Lambda_\pm(t) = \lambda_\pm w(t), \qquad B_t \perp D_t.$$

Proposition A.5 (Bessel slack posterior and closure under time-rescaling)**.** *Fix $t \in (0, 1]$ and let $d_t = X_t - X_0 = B_t - D_t$. Conditional on d_t, the pair (B_t, D_t) lies on the lattice*

$$(B_t, D_t) = \begin{cases} (m + d_t,\, m), & d_t \geq 0, \\ (m,\, m + |d_t|), & d_t < 0, \end{cases} \qquad m \in \mathbb{N},$$

and the induced posterior pmf of the slack $M_t = \min(B_t, D_t)$ is

$$\pi_t(m \mid d_t) = \mathbb{P}(M_t = m \mid d_t) \;\propto\; \frac{(\Lambda_+(t)\Lambda_-(t))^m}{(m + |d_t|)!\, m!}, \qquad m = 0, 1, 2, \ldots \tag{11}$$

with normalizer

$$Z_t(d_t) = \big(\Lambda_+(t)\Lambda_-(t)\big)^{-|d_t|/2} I_{|d_t|}\Big(2\sqrt{\Lambda_+(t)\Lambda_-(t)}\Big),$$

where I_ν is the modified Bessel function of the first kind.

*Moreover—**closure under time-rescaling**: for any $0 < s < t$,*

$$\pi_s(\,\cdot \mid d_s) \text{ has the same functional form as } \pi_t(\,\cdot \mid d_t),$$

with parameters obtained by replacing $(\Lambda_+(t), \Lambda_-(t))$ by $(\Lambda_+(s), \Lambda_-(s))$. Since $\Lambda_\pm(s) = \lambda_\pm w(s)$, this closure depends only on the rescaling of $w(\cdot)$ and not on any other properties of the process.

Proof. We treat $d_t \geq 0$, the other case being symmetric. If $d_t = B_t - D_t \geq 0$ then $B_t = D_t + d_t$ and thus

$$(B_t, D_t) = (m + d_t, m) \quad \Longleftrightarrow \quad M_t = m.$$

The joint pmf at $(m + d_t, m)$ is

$$\mathbb{P}(B_t = m + d_t,\, D_t = m) = e^{-(\Lambda_+(t)+\Lambda_-(t))} \frac{\Lambda_+(t)^{m+d_t}}{(m + d_t)!} \frac{\Lambda_-(t)^m}{m!}.$$

Conditioning on d_t amounts to normalizing over m:

$$\pi_t(m \mid d_t) = \frac{\Lambda_+(t)^{m+d_t}\Lambda_-(t)^m / \big((m + d_t)!\, m!\big)}{\sum_{r=0}^{\infty} \Lambda_+(t)^{r+d_t}\Lambda_-(t)^r / \big((r + d_t)!\, r!\big)}.$$

Factor $\Lambda_+(t)^{d_t}$ from numerator and denominator to obtain

$$\pi_t(m \mid d_t) = \frac{(\Lambda_+(t)\Lambda_-(t))^m}{(m + d_t)!\, m!} \Bigg/ \sum_{r=0}^{\infty} \frac{(\Lambda_+(t)\Lambda_-(t))^r}{(r + d_t)!\, r!},$$

which is exactly equation 11. The denominator is the standard Skellam normalizer, given by the stated Bessel expression.

For closure, note that for any $s < t$ we have independent thinning

$$B_s \sim \mathrm{Poi}(\Lambda_+(s)), \qquad D_s \sim \mathrm{Poi}(\Lambda_-(s)), \qquad \Lambda_\pm(s) = \lambda_\pm w(s).$$

Repeating the same conditioning argument with (B_s, D_s) and d_s yields the same functional family as equation 11, with parameters $(\Lambda_+(s), \Lambda_-(s))$ replacing $(\Lambda_+(t), \Lambda_-(t))$. Thus π_t is closed under the time-rescaling $w(\cdot)$. □

Mixing Over the Slack

This proposition shows that the slack posterior depends on the state only through the observable displacement $d_t = X_t - X_0$ and the posterior family $\pi_t(\cdot \mid d_t)$ is closed under passing to earlier times $s < t$.

Using these properties, combined with the composition in a fixed slack, we can prove the general Markovanity after mixing over the slack. We now mix the fixed–slack bridges over the Bessel slack posterior to obtain an observable bridge that depends only on the counts X_t and satisfies the same bridge consistency identity as in equation 1.

Recall the fixed–slack bridge kernels from Theorem A.4:

$$K^{(m)}_{a|0,b}(x_a \mid x_0, x_b) = \mathbb{P}\big(X_a = x_a \,\big|\, X_0 = x_0, X_b = x_b, M_1 = m\big), \qquad 0 \le a \le b \le 1.$$

For each $m \in \mathbb{N}$ these satisfy fixed–slack bridge consistency: for all $0 < s < t < 1$,

$$K^{(m)}_{s|0,1}(x_s \mid x_0, x_1) = \sum_{x_t} K^{(m)}_{s|0,t}(x_s \mid x_0, x_t)\, K^{(m)}_{t|0,1}(x_t \mid x_0, x_1). \tag{12}$$

At the terminal time 1, define the observable $(0, 1)$ bridge by

$$K_{s|0,1}(x_s \mid x_0, x_1) \;=\; \sum_{m=0}^{\infty} \pi_1(m \mid d_1)\, K^{(m)}_{s|0,1}(x_s \mid x_0, x_1), \qquad d_1 = x_1 - x_0.$$

At an intermediate time t, define the observable $(0, t)$ bridge by

$$K_{s|0,t}(x_s \mid x_0, x_t) \;=\; \sum_{m=0}^{\infty} \pi_t(m \mid d_t)\, K^{(m)}_{s|0,t}(x_s \mid x_0, x_t), \qquad d_t = x_t - x_0,$$

and similarly

$$K_{t|0,1}(x_t \mid x_0, x_1) \;=\; \sum_{m=0}^{\infty} \pi_1(m \mid d_1)\, K^{(m)}_{t|0,1}(x_t \mid x_0, x_1).$$

Theorem A.6 (Bridge consistency of the observable count bridge)**.** *Under the Poisson birth–death reference and Bessel slack mixing, the observable bridge kernels satisfy the discrete form of equation 1: for all* $0 < s < t < 1$ *and endpoints* x_0, x_1*,*

$$K_{s|0,1}(x_s \mid x_0, x_1) = \sum_{x_t} K_{s|0,t}(x_s \mid x_0, x_t)\, K_{t|0,1}(x_t \mid x_0, x_1). \tag{13}$$

Proof. Fix x_0, x_1 and a bounded test function φ on the state space. We prove equation 13 in weak form by comparing $\mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_1 = x_1]$ computed in two ways.

First, by definition of $K_{s|0,1}$ and the slack posterior at time 1,

$$\mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_1 = x_1] = \sum_{m=0}^{\infty} \pi_1(m \mid d_1)\, \mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_1 = x_1, M = m],$$

with $d_1 = x_1 - x_0$. For each fixed m, the bridge with slack $M = m$ satisfies the fixed–slack bridge consistency equation 12, so

$$\begin{aligned}&\mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_1 = x_1, M = m] \\ &\quad = \sum_{x_t} \mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_t = x_t, M = m]\, K^{(m)}_{t|0,1}(x_t \mid x_0, x_1).\end{aligned}$$

Substitute into the previous display and interchange the sums over x_t and m:

$$\begin{aligned}&\mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_1 = x_1] \\ &\quad = \sum_{x_t} \sum_{m=0}^{\infty} \pi_1(m \mid d_1)\, \mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_t = x_t, M = m]\, K^{(m)}_{t|0,1}(x_t \mid x_0, x_1).\end{aligned}$$

Now condition on the intermediate state $X_t = x_t$ and use Bayes' rule. By Proposition A.5, the posterior over M given $(X_0 = x_0, X_t = x_t)$ is exactly $\pi_t(\cdot \mid d_t)$ with $d_t = x_t - x_0$, i.e.

$$\pi_t(m \mid d_t) = \mathbb{P}(M = m \mid X_0 = x_0, X_t = x_t).$$

Hence we can rewrite the inner sum over m as

$$\sum_{m=0}^{\infty} \pi_t(m \mid d_t)\, \mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_t = x_t, M = m] = \mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_t = x_t],$$

and the definition of $K_{s|0,t}$ then gives

$$\mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_t = x_t] = \sum_{x_s} \varphi(x_s)\, K_{s|0,t}(x_s \mid x_0, x_t).$$

Putting everything together,

$$\begin{aligned}\mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_1 = x_1] &= \sum_{x_t} \Big(\sum_{x_s} \varphi(x_s)\, K_{s|0,t}(x_s \mid x_0, x_t) \Big) K_{t|0,1}(x_t \mid x_0, x_1) \\ &= \sum_{x_s} \varphi(x_s) \Big(\sum_{x_t} K_{s|0,t}(x_s \mid x_0, x_t)\, K_{t|0,1}(x_t \mid x_0, x_1) \Big).\end{aligned}$$

On the other hand, by the definition of $K_{s|0,1}$,

$$\mathbb{E}[\varphi(X_s) \mid X_0 = x_0, X_1 = x_1] = \sum_{x_s} \varphi(x_s)\, K_{s|0,1}(x_s \mid x_0, x_1).$$

Since φ is arbitrary, the coefficients of $\varphi(x_s)$ must agree for all x_s, which yields equation 13. □

This theorem shows that after mixing over the Bessel slack posterior at each time, the observable birth–death bridge has the same bridge consistency property equation 1 as the Gaussian diffusion bridge in Proposition 2.1.

A.2 Link with Schrödinger Bridges

We now justify the Schrödinger-bridge interpretation stated in Sec. 3. Recall the unconditional birth–death construction

$$X_t \;=\; X_0 + B_t - D_t,$$

where $(B_t)_{t\in[0,1]}$ and $(D_t)_{t\in[0,1]}$ are independent Poisson processes with cumulative intensities $\Lambda_+(t) = \lambda_+ w(t),\ \Lambda_-(t) = \lambda_- w(t)$, and $w(0) = 0, w(1) = 1$. Let $\kappa = \sqrt{\lambda_+ \lambda_-}$ denote the (scalar) jump intensity. Then for each $x_0 \in \mathcal{X}$, the forward kernel at time $t = 1$ has Skellam pmf

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

where I_ν is the modified Bessel function of the first kind.

Let

$$p^{\kappa}_{\mathrm{ref}}(x_0, x_1) \;=\; p_0(x_0)\, K^{\kappa}_{1|0}(x_1 \mid x_0)$$

be the *reference joint law* of (X_0, X_1) induced by the birth–death process and the data prior p_0. Over the space of couplings

$$\mathcal{C}(p_0, p_1) = \big\{ C \text{ on } \mathcal{X} \times \mathcal{X} : C(\cdot, \mathcal{X}) = p_0,\ C(\mathcal{X}, \cdot) = p_1 \big\},$$

we consider the entropic projection

$$C_\kappa \in \arg\min_{C \in \mathcal{C}(p_0, p_1)} \mathrm{KL}\big(C \,\|\, p^{\kappa}_{\mathrm{ref}}\big).$$

Fix any $C \in \mathcal{C}(p_0, p_1)$ and write expectations with respect to C as $\mathbb{E}_C[\cdot]$. By definition,

$$\begin{aligned}\mathrm{KL}\big(C \,\|\, p^{\kappa}_{\mathrm{ref}}\big) &= \mathbb{E}_C\left[\log \frac{dC}{dp^{\kappa}_{\mathrm{ref}}}(X_0, X_1)\right] \\ &= -H(C) - \mathbb{E}_C\big[\log K^{\kappa}_{1|0}(X_1 \mid X_0)\big] + C_\kappa(p_0, p_1),\end{aligned}$$

where $H(C)$ is the Shannon entropy of C and $C_\kappa(p_0, p_1)$ collects all terms depending only on the marginals (including $\Lambda_\pm(1)$ and $\mathbb{E}_C[X_1 - X_0]$, which are fixed across all couplings with marginals p_0, p_1).

Using the explicit Skellam form, we can isolate the Bessel contribution:

$$\mathrm{KL}\big(C \,\|\, p_{\mathrm{ref}}^{\kappa}\big) = C_\kappa(p_0, p_1) - \mathbb{E}_C\big[\log I_{|X_1 - X_0|}(2\kappa)\big] - H(C).$$

To study the limits $\kappa \to \infty$ and $\kappa \to 0^+$, we use standard asymptotics for I_ν (NIS, 2025, §10.41(ii)):

$$I_\nu(z) = \frac{e^z}{\sqrt{2\pi z}}\,(1 + o(1)) \quad \text{as } z \to \infty, \qquad I_\nu(z) = \frac{(z/2)^\nu}{\Gamma(\nu{+}1)}\,(1 + o(1)) \quad \text{as } z \to 0^+.$$

High-noise limit $\kappa \to \infty$. For fixed ν, the large-z expansion shows that $\log I_\nu(2\kappa)$ depends on ν only through $O(1/\kappa)$ terms. Hence, as $\kappa \to \infty$,

$$\mathbb{E}_C\big[\log I_{|X_1 - X_0|}(2\kappa)\big] = C'_\kappa(p_0, p_1) + o_\kappa(1),$$

where $C'_\kappa(p_0, p_1)$ does not depend on the choice of coupling C. It follows that

$$\mathrm{KL}\big(C \,\|\, p_{\mathrm{ref}}^{\kappa}\big) = C - H(C) + o_\kappa(1), \qquad \kappa \to \infty,$$

for some constant C that is independent of C. Maximizing $H(C)$ over $\mathcal{C}(p_0, p_1)$ yields the independent coupling $p_0 \otimes p_1$, so in the high-noise limit $C_\kappa \to p_0 \otimes p_1$.

Low-noise limit $\kappa \to 0^+$. For small z, we have

$$\log I_\nu(2\kappa) = \nu \log\Big(\tfrac{2}{\kappa}\Big) + O(1) \quad \text{as } \kappa \to 0^+,$$

uniformly for integer ν in any fixed finite range. Applying this with $\nu = |X_1 - X_0|$ gives

$$-\mathbb{E}_C\big[\log I_{|X_1 - X_0|}(2\kappa)\big] = \log\Big(\tfrac{2}{\kappa}\Big)\,\mathbb{E}_C|X_1 - X_0| + O(1), \quad \kappa \to 0^+.$$

Substituting into the expression for the KL divergence, we obtain

$$\mathrm{KL}\big(C \,\|\, p_{\mathrm{ref}}^{\kappa}\big) = C + \log\Big(\tfrac{2}{\kappa}\Big)\,\mathbb{E}_C|X_1 - X_0| - H(C) + o_\kappa(1), \qquad \kappa \to 0^+,$$

for some constant C independent of C.

Thus, in the low-noise limit the dominant term in $\mathrm{KL}(C \,\|\, p_{\mathrm{ref}}^{\kappa})$ is proportional to $\mathbb{E}_C|X_1 - X_0|$; minimizing the KL over $\mathcal{C}(p_0, p_1)$ therefore recovers discrete optimal transport with cost $|x_1 - x_0|$, while the entropy term $-H(C)$ corresponds to the usual entropic regularization. This proves the characterization stated in Sec. 3.

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 $C_{X_{g0}}(t) = F(t; \psi_g)$ 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:*

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

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,*

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

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

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

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):

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

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 $\{X_{g0}^{(G)} : 1 \leq g \leq G\}$ 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 $k \geq 3$. Thus:*

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

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

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 run up to time τ. 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.

[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.

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,

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

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

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

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 Projection and Rounding Algorithms

Group rescaling. Algorithm 5 rescales item-level nonnegative values $\{x_b\}_{b=1}^B$ so that each group G_g attains a prescribed aggregate C_g. The procedure is linear-time in the number of items and linear in memory in the number of groups:

$$T = O(B), \qquad M_{\text{extra}} = O(G),$$

and is applied in parallel to each feature dimension.

Algorithm 5 Group Rescaling to Match Aggregates (scalar form)

Require: item values $x_b \geq 0$ for $b = 1, \ldots, B$; groups $G_1, \ldots, G_G$; targets $C_g \geq 0$
Ensure: $y_b \geq 0$ with $\sum_{b \in G_g} y_b = C_g$ for all g

for $g = 1, \ldots, G$ **do**
  $S_g \leftarrow \sum_{b \in G_g} x_b$
  if $S_g > 0$ **then**
    for $b \in G_g$ **do**
      $y_b \leftarrow x_b \, C_g / S_g$ ▷ proportional rescaling
    end for
  else
    for $b \in G_g$ **do**
      $y_b \leftarrow C_g / |G_g|$ ▷ uniform split
    end for
  end if
end for
return $\{y_b\}$

Randomized rounding. Algorithm 6 independently rounds a real value to the nearest integers, preserving the value in expectation. It runs in constant time and memory per entry, so applying it over B items has

$$T = O(B), \qquad M_{\text{extra}} = O(1).$$

Algorithm 6 Randomized Rounding (scalar form)

Require: real value $x \geq 0$
Ensure: integer $y \in \mathbb{Z}_{\geq 0}$

$a \leftarrow \lfloor x \rfloor$
$r \leftarrow x - a$
sample $U \sim \mathrm{Unif}[0, 1]$
if $U < r$ **then**
  $y \leftarrow a + 1$
else
  $y \leftarrow a$
end if
return y

Groupwise exact rounding. Algorithm 7 converts rescaled real values $\{x_b\}$ to integers $\{y_b\}$ while *exactly* preserving each group sum: $\sum_{b \in G_g} y_b = C_g$. Each y_b differs from x_b by at most 1. The only expensive step is weighted sampling without replacement inside each group. With Gumbel–Top-k sampling the worst-case complexity is

$$T = O\Big(\sum_{g=1}^{G} |G_g| \log |G_g| \Big), \qquad M_{\text{extra}} = O(B),$$

and the algorithm is again applied independently over coordinates.

Algorithm 7 Groupwise Randomized Rounding with Exact Aggregates (scalar form)

Require: rescaled $x_b \geq 0$, groups $G_1, \ldots, G_G$, integer targets C_g
Ensure: integers $y_b \geq 0$ with $\sum_{b \in G_g} y_b = C_g$ and $|y_b - x_b| \leq 1$

for $g = 1, \ldots, G$ **do** ▷ work inside group G_g
  for $b \in G_g$ **do**
    $a_b \leftarrow \lfloor x_b \rfloor$
    $r_b \leftarrow x_b - a_b$
  end for
  $A_g \leftarrow \sum_{b \in G_g} a_b$
  $S_g \leftarrow C_g - A_g$ ▷ # of increments required
  if $S_g = 0$ **then**
    for $b \in G_g$ **do**
      $y_b \leftarrow a_b$
    end for
  else
    sample a subset $S \subseteq G_g$ of size S_g
    without replacement with weights $\propto r_b$
    for $b \in G_g$ **do**
      if $b \in S$ **then**
        $y_b \leftarrow a_b + 1$
      else
        $y_b \leftarrow a_b$
      end if
    end for
  end if
end for
return $\{y_b\}$

D 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 $\sqrt{\Lambda_+ \Lambda_-} = 32$.

D.1 DISCRETE 8-GAUSSIANS TO 2-MOONS

D.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.

D.1.2 Results

We present a visualization of the learned trajectories in Fig. 2 and the full details in Table 6. 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 6: Discrete Moons Results: Noise → 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

We also run the Count Bridge across different noise levels, here we actually find that our default of $\lambda_+ = \lambda_- = 32$ is not optimized, so all results can be considered lower bounds on our performance.

Table 7: Count Bridge (Energy Score) Results Across Different $\lambda_+ = \lambda_-$ Values

$\lambda_+ = \lambda_-$	MMD	W_2	Energy
0	**0.0038 ± 0.0012**	0.0046 ± 0.0002	**0.0075 ± 0.0011**
8	0.0039 ± 0.0015	**0.0045 ± 0.0006**	0.0080 ± 0.0022
16	0.0049 ± 0.0003	0.0049 ± 0.0003	0.0095 ± 0.0004
32	0.0052 ± 0.0024	0.0055 ± 0.0015	0.011 ± 0.006
256	0.0064 ± 0.0020	0.0063 ± 0.0009	0.015 ± 0.004

D.1.3 Noise to 2-Moons for Diffusion Comparisons

Here we compare against a standard Gaussian DDIM model Song et al. (2020) and Discrete Diffusion as in Shi et al. (2024). Since these models go from noise to a target distribution, we cannot do the 8-Gaussians to 2-Moons task, so we simply target the 2-Moons. This makes the task substantially easier. For Count Bridges we use the same results from the previous table (the more difficult task, but with comparable scores at the endpoint).

Table 8: Discrete Moons Results: Eight Gaussians → Two Moons

Method	MMD	W_2	Energy
Count Bridge (ES)	**0.0044 ± 0.0018**	**0.0052 ± 0.0007**	**0.0098 ± 0.0029**
Gaussian Diffusion	0.024 ± 0.009	0.017 ± 0.006	0.118 ± 0.064
Discrete Diffusion	0.017 ± 0.010	0.013 ± 0.006	0.072 ± 0.055

D.2 Low-Rank Gaussian Mixture

D.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.

D.2.2 Results

The central scaling results are presented visually in Fig. 3. Here we also present the full experimental details in Table 9.

D.3 Deconvolution Gaussian Mixture Dataset

We extend the low-rank Gaussian mixture task (Appendix D.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).

Table 9: 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**

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 10, 11, 12.

Table 10: 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 11: 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 12: 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

To investigate the importance of different rounding approaches in our deconvolution implementation we run an ablation study where we substitute our preferred exact rounding approach for two alternatives: first a simple deterministic rounding, and second a randomized rounding (where we simply add zero or one with probability of the decimal value). Deterministic rounding can lead to arbitrarily incorrect results, randomized rounding preserves the expectation, and our exact approach will ensure we exactly match the target aggregate value. We find that although our exact approach is superior across three replicates of the $n = 128, \alpha = 1$ setting the results are very close particularly for the randomized rounding. We believe this could justify substituting the randomized approach since it is simpler, although in different regimes this may matter more or less.

Table 13: Performance comparison of different rounding methods with standard errors.

Method	MMD	W_2	EMD
`exact`	$\mathbf{0.037 \pm 0.005}$	$\mathbf{0.050 \pm 0.008}$	$\mathbf{2.24 \pm 0.58}$
`randomized`	0.038 ± 0.000	0.051 ± 0.007	2.33 ± 0.43
`round`	0.038 ± 0.003	0.052 ± 0.007	2.32 ± 0.44

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

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

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:

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

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

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

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 $N_{\text{proj}}{=}2$ self–attention layers (sequence–first API) with residual+LayerNorm are applied:

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

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

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

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

[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.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 $p_{\text{agg}}{=}0.1$ 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 $p_{\text{mask}}{=}0.1$.

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.

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**

We also analyze the unit-level profiles of the deconvolved transciptomes. We aggregate the nucleotide level transcriptomes up to the gene level by computing the maximum count over the gene profile, enabling us to generate a count matrix from our deconvolved profile. We then assign the cell types and plot the UMAP of the held-out ground truth vs the deconvolved transcriptomic profiles. We can see that the deconvolved profiles are realistically clustered mirroring the ground truth 5.

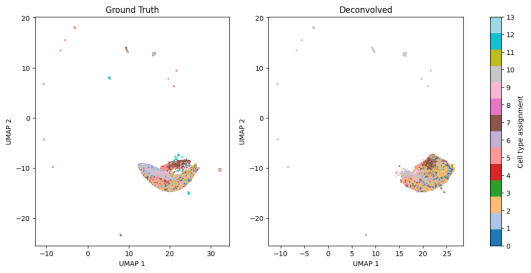


Figure 5: Ground Truth vs Deconvovled UMAP for nucleotide level bulk deconvolution, aggregated to the gene level and hued by cell type (as assigned by Yazar et al. (2022)).

F Spatial transcriptomic deconvolution

F.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.

F.2 Architecture, Training, and Sampling

F.2.1 Inputs and tokenization

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

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

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.

F.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`:

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

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

F.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):

$$\mathcal{L}(\theta) \;=\; -\mathbb{E}_{t,(X_0,X_t)}\Big[\, S_\rho\big(q_\theta(\cdot \mid X^{\text{cnt}}_t, I_t, t, \varepsilon, y),\, X^{\text{cnt}}_0\big)\,\Big],$$

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 $\tilde{X}_0 \sim q_\theta(\cdot \mid x^{\text{cnt}}_t, I_t, t, \varepsilon, y)$ and update $x_{t-\Delta}$ using the exact binomial–hypergeometric count-bridge kernel (Prop. 3.1).

F.2.4 IMPLEMENTATION SPECIFICS

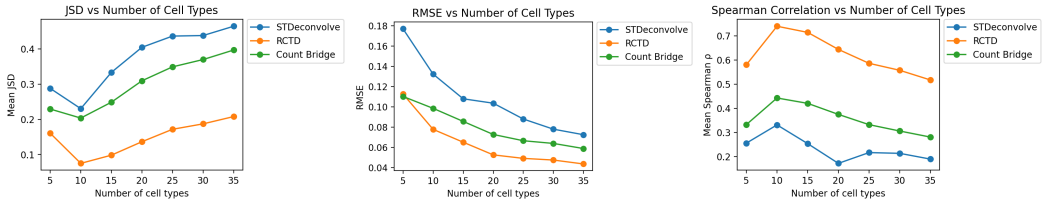
- **Patchification.** `PatchEmbed` uses patch size P on I_t (channels C), producing $N_{\text{img}} = (H/P)(W/P)$ tokens of width d. `CountPatchEmbedding` projects D_c-dimensional counts to N_{cnt} tokens of width d with learned positional embeddings.
- **Auxiliary tokens.** Time token: `timestep_embedding` 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.

F.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.

F.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 $x_{t_{k-1}}$.



(a) (b) (c)

Figure 6: Performance of RCTD, STDeconvolve and Count Bridge on MERFISH deconvolution across number of cell types using (a) Jensen Shannon Divergence (JSD), (B) RMSE and (C) Spearman Correlation

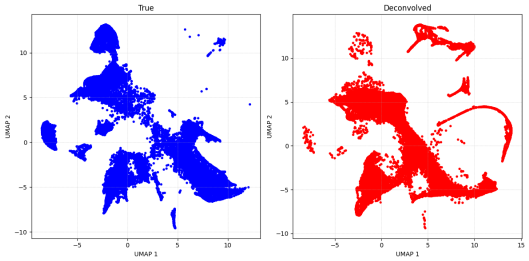


Figure 7: UMAP of true vs deconvolved cell profiles using the Count Bridge.

F.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. 14. For evaluation, we use Jensen shannon Divergence (JSD) and RMSE metrics as described in Li et al. (2023). 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.

Method	JSD	RMSE	Spearman
STDeconvolve	0.288	0.177	0.255
RCTD	0.161	0.113	0.580
Count Bridge	0.229	0.110	0.332

Table 14: Cell-type deconvolution error for spatial transcriptomic data. Note that RCTD requires a single-cell reference dataset for deconvolution.

In Fig. 6, 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.

Inspection of unit-level data generated by Count Bridges In the previous section, we have shown through quantitative metrics that Count Bridges outperform alternatives for reconstructing unit-level gene expression vectors from spot-level aggregates. We next aim to evaluate the extent to which reconstructed gene expression profiles are biologically meaningful. We do this by performing conventional single-cell transcriptomic analysis on the synthetic unit-level expression vectors generated by Count Bridges.

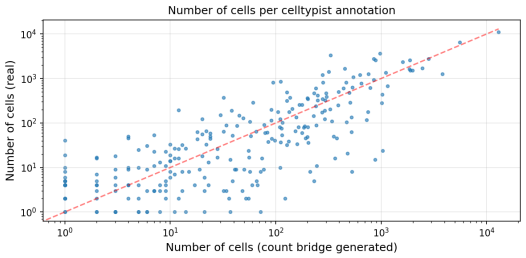


Figure 8: Abundance of putative cell types (automatically annotated by celltypist) in synthetic unit-level data generated by Count Bridges through deconvolution, vs. abundance of putative cell types in true unit-level data.

First, we plot the umap of the raw expression profiles of the true and generated data in 7. We can clearly see that, without ever seeing unit-level data, our deconvolution approach clearly learns a realistic distribution of cells. This is visually clear in the distribution of the UMAP.

Second we will perform cell-type annotation. We preprocess by normalizing counts to 10^4 per cell (row-normalizing), followed by a log-transform, then annotate cell types using celltypist (Domínguez Conde et al., 2022). From this process, we identify 268 putative cell types in the synthetic unit-level data. The same pipeline, when applied to the real single-cell level measurements, identifies 278 putative cell types. And as shown in Figure 8, the cell type abundances inferred by Count Bridges (without access to unit level data) closely align with the cell type abundances observed in the true unit level data.

Deconvolution of a 10X Visium dataset We next evaluate the deconvolution of spots in a real world 10X Visium dataset profiling the mouse brain. As ground truth is unavailable in this setting, we validate model predictions by assessing the extent to which the synthetic deconvolved data reflects known biology.

We use the 10X Visium fluorescence dataset distributed by Palla et al. (2022), which profiles a coronal section of a mouse brain. To demonstrate the generalization capabilities of Count Bridges, we apply the model trained on MERFISH data directly to this Visium dataset without retraining.

In order to correct batch effects and align dimensionality between datasets, we employ a moment-matching procedure. For each gene in the MERFISH data, we compute the mean and variance of expression and identify the gene in the 10X Visium data that most closely matches these moments. We then map the 10X Visium count matrices to the MERFISH feature space by subsetting to these matched genes.

We deconvolve the 10X Visium data using Count Bridges with a spot-level mean constraint. To evaluate prediction quality, we use a standard single-cell analysis pipeline (as described above) and determine putative cell type annotations using Celltypist (Domínguez Conde et al., 2022). Celltypist identifies 146 putative cell types, suggesting that the synthetic unit-level data recapitulates a significant degree of cell-to-cell variation. Furthermore, the recovered cell types are biologically consistent: the most abundant identified cell type is the oligodendrocyte, which matches the most abundant annotation in the MERFISH mouse brain dataset described above.