JUMPING UP AND DOWN: DENOISER DIFFUSION MODELS FOR DISCRETE ORDINAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are highly developed in continuous spaces for image and video domains. Recently, major advances have been made for discrete diffusion models for categorical data, specifically in the language domain. In contrast, diffusion models for discrete integer-valued data are less developed, despite the prevalence of this modality, ranging from images and music to gene counts. We introduce Jumping Up and Down (JUD)—a new family of denoiser-based diffusion models for discrete ordinal data. This is the first family of diffusion models for ordinal data which centers around training denoisers, which at the same time allows for bi-directional (up and down) perturbations of the data. The simplicity of the training objective, combined with the flexibility of bi-directional perturbations, leads us to obtain competitive results across different data modalities.

1 INTRODUCTION

Diffusion models for discrete ordinal data. The success of diffusion models in continuous domains such as image and video is driving an explosion of activity in applying diffusion models for discrete data. The vast majority of diffusion models for discrete data focus on language where the data is *categorical*, that is, a data modality with no natural order. In contrast, in many applications the discrete data also contains a natural order, i.e., the data is *ordinal* as its distribution is over (subsets of) $\mathbb{N}^D$ or $\mathbb{Z}^D$. This modality naturally shows up in images (8-bit digital grayscale from 0 to 255), music (MIDI note numbers from 0 to 127), tabular data (amount in dollars of debt/credit), and in scientific applications ranging from biology (gene expression counts in cells) to chemistry (number of atoms in molecules). In addition to the aforementioned applications of discrete ordinal data, this modality is also interesting from the theoretical perspective as it lies in some sense between categorical and continuous data: On the one hand, the data is discrete as in categorical data, while on the other hand the data has an intrinsic order as in continuous data in $\mathbb{R}^D$. Despite the importance of discrete ordinal data, the theory and practice of diffusion models for this modality are still developing. In this paper we contribute to both.

Denoisers. The essence of diffusion models is learning clean data from progressively noisy versions of the same data point. In continuous space, the noise is often taken to be Gaussian with the level of noise corresponding to the variance of the Gaussian. The choice of Gaussian noise is very convenient since it provides a clean relation between *denoising* and *sampling*. First, one learns the **data denoiser**,

$$\mathrm{m}_{\text{data}}(\mathbf{x}, t) := \mathbb{E}[\mathbf{x}_{\text{data}} \,|\, \texttt{CorruptData}(\mathbf{x}_{\text{data}}, t) = \mathbf{x}], \qquad \mathbf{x}_{\text{data}} \sim p_{\text{data}}, \tag{1}$$

where $\texttt{CorruptData}(\mathbf{x}_{\text{data}}, t)$ can be, for example, adding to $\mathbf{x}_{\text{data}}$ Gaussian noise with variance t. Second, the data denoiser m_{data} can be directly used for sampling via **Tweedie's formula**, which connects the denoiser m_{data} to the *score* $\nabla \log p_t$, where p_t is the distribution of t-level corrupted data. The validity of Tweedie's formula is due to the use of *Gaussian* noise when corrupting the data. The advantages of formulating a training objective based on the denoiser were recently emphasized in Li & He (2026); Hu et al. (2026). In contrast to the continuous setting, the absence of the Gaussian distribution in the discrete setting necessitates the development of a different concept of "data corruption". For categorical data, the two prevailing choices for noise are uniform (choosing a token uniformly at random) or masking (adding an extra graveyard token), and the level of noise corresponds to the rate at which a token is flipped to a random or masked token. The lack of ordinal

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

structure in categorical data has led to deviation from training a denoiser as in Equation (1), since the notion of average (expectation) does not make sense in such generality[1]. Instead, the common training objectives in diffusion models for categorical data are either *discrete scores* (Meng et al., 2022; Lou et al., 2024), which are the analogues of $\nabla \log p_t$ in the discrete setting, or the *rates* of the generative process (Campbell et al., 2022; Holderrieth et al., 2025). But for ordinal data, the definition of a denoiser analogous to (1) does make sense, and learning the denoiser has a number of advantages:

- The denoiser can be trained with a large family of losses (Bregman divergences), including the simple mean-square error loss. Alternatively, the denoiser can be learned by training cross-entropy objectives for the *marginal* posteriors; see Section 2.
- Techniques from continuous diffusion models, which are denoiser-based, can in principle be adapted to the discrete ordinal setting: EDM scaling (Karras et al., 2022), Distillation via Moment Matching (Salimans et al., 2024), dual score matching (Guth et al., 2025), blind denoisers (Sun et al., 2025; Wang & Du, 2025; Kadkhodaie et al., 2026), etc. See Section 3 for our use of the EDM scaling.

In Figure 1 we compare various noising processes and methods for discrete diffusion models that are trained using the same set of $20,000$ data samples from a bi-modal Negative Binomial Mixture distribution; see Section F for more details. As we will demonstrate in this work, these improvements show up not only on toy examples but on real-world datasets as well.

Jumping up and down In continuous spaces, the Gaussian noise blurs the data in all directions. For ordinal data, a data point in $\mathbb{N}$ or $\mathbb{Z}$ can be perturbed either up, or down, or both. As we will see below, uni-directional perturbations (either just up or just down) give rise to denoiser-based training and sampling (cf. Section D). That is, provided a suitable choice of up-perturbation, or down-perturbation, we can train an appropriate denoiser m_{data} and then use it to sample from the data distribution. But perturbing the data in only one direction can lead to inaccurate samplers. This is because once the sampler over/undershoots, due to inevitable approximation errors, the process cannot "undo" prior mistakes and correct the trajectory evolution. This is evident in Figure 1 when comparing Binomial-Poisson (one of our up-and-down methods), Binomial (only down) and Poisson (only up). Hence, our goal is to find perturbations and samplers which allow for up *and* down jumps, while at the same time being *denoiser-based*. This is a non-trivial task, as the up and down perturbations can interact in arbitrarily complex ways, which was not accomplished in previous works; for space reasons, related works are discussed in Section I. This is precisely the goal that we will achieve in this paper.

Our contributions

- We develop a novel framework of **denoiser-based** diffusion models (JUD) for discrete ordinal data on $\mathbb{N}^D$ and $\mathbb{Z}^D$. Our key innovation is to train **latent** denoisers which predict **partially noised data**. Our main result (Theorem 2.1) shows that the latent denoisers are sufficient to sample from the data distribution via a stochastic process which allows for **Up-and-Down** moves.
- We demonstrate the effectiveness of this framework on synthetic data and real-world data in domains such as images (CIFAR-10), and reasoning (Sudoku completion). In particular, we obtain state-of-the-art FID on CIFAR-10 among discrete diffusion models.

2 JUMPING UP AND DOWN (JUD)

Continuous diffusion models rely on *Tweedie's formula*. For $\mathbf{x}_{\text{data}} \sim p_{\text{data}}$ on $\mathbb{R}^D$, add independent Gaussian noise with a continuous schedule $\sigma_t > 0$,

$$Z_t := \mathbf{x}_{\text{data}} + G_t, \qquad G_t \sim \mathcal{N}(0, v_t \,\mathrm{Id}_D), \qquad v_t := \int_0^t \sigma_u^2 \,\mathrm{d}u, \tag{2}$$

[1]The term "denoiser" is sometimes used in the literature on language diffusion models to refer to the prediction of the log probabilities over tokens. Here we use the term "denoiser" strictly for the prediction of the data point itself, as in Equation (1). Below we explain the advantages of directly training the latter.

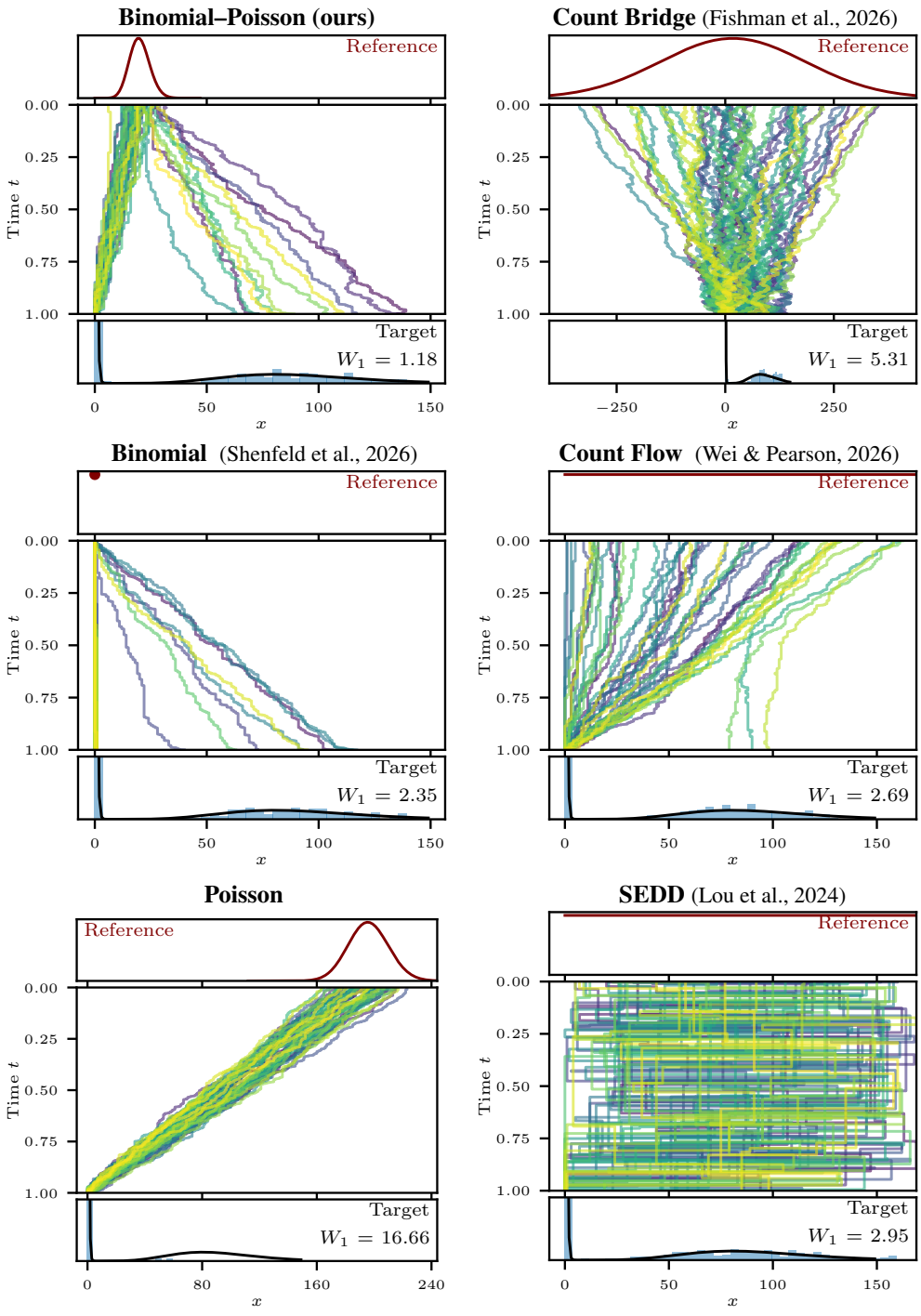

Figure 1: Sample paths for several noising processes in discrete diffusion models and the resulting histogram of generated samples (blue) relative to the true data distribution (black) *Negative Binomial Mixture*. We observe closest agreement in W_1 distance to the data with the **Binomial-Poisson** process proposed in this work. See Section F for details.

and let p_t be the law of Z_t, so $p_0 = p_{\text{data}}$. For $t > 0$, Tweedie's formula relates the score of p_t to the *data denoiser* $\mathrm{m}_{\text{data}}(\mathbf{x}, t) := \mathbb{E}[\mathbf{x}_{\text{data}} | Z_t = \mathbf{x}]$:

$$v_t \nabla \log p_t(\mathbf{x}) = \mathrm{m}_{\text{data}}(\mathbf{x}, t) - \mathbf{x}. \tag{3}$$

The denoiser can be trained by regression (or any Bregman divergence), providing the score for sampling. For a final time τ, the stochastic differential equation (SDE) in reverse time $s = \tau - t$,

$$\mathrm{d}\mathbf{Z}_s = \sigma_{\tau-s}^2 \nabla \log p_{\tau-s}(\mathbf{Z}_s)\mathrm{d}s + \sigma_{\tau-s}\mathrm{d}\mathbf{W}_s, \tag{4}$$

with $\mathbf{Z}_0 \sim p_\tau$ and standard D-dimensional Brownian motion $(\mathbf{W}_s)$, has marginals $\mathbf{Z}_s \sim p_{\tau-s}$ and hence $\mathbf{Z}_\tau \sim p_{\text{data}}$ (Song et al., 2021b).

Consider now the setting where p_{data} is a distribution over $\mathbb{N}^D$ or $\mathbb{Z}^D$. A Gaussian perturbation will lead to non-integer values so a different type of noise must be considered. In this paper we focus on the generative modeling framework of continuous-time Markov-chains (CTMCs) (Campbell et al., 2022) for discrete data.

In order to have an analogue of Equation (3) in our setting, we will consider either *Poisson* noise or *Binomial* noise. In detail, when p_{data} is supported over $\mathbb{N}^D$ we perturb a data point $\mathbf{x}_{\text{data}} \sim p_{\text{data}}$ by

$$X_t := \mathbf{x}_{\text{data}} - \mathrm{L}_t + \mathrm{N}_t, \qquad \text{with} \qquad \mathrm{L}_t \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}, \mathrm{d}_t), \quad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{5}$$

where N_t is independent of $(\mathbf{x}_{\text{data}}, \mathrm{L}_t)$. Here, $\mathrm{Binomial}(\mathbf{x}_{\text{data}}, \mathrm{d}_t)$ is a D-dimensional product Binomial distribution whose ith coordinate has $\mathbf{x}_{\text{data}}^i$ trials and probability of success $\mathrm{d}_t^i \in [0, 1]$, and $\mathrm{Poisson}(\mathrm{u}_t)$ is the D-dimensional product Poisson distribution with a vector of parameters $\mathrm{u}_t \in \mathbb{R}_{\geq 0}^D$. Taking τ to be the final time, and choosing $\mathrm{d}_0 = 0, \mathrm{d}_\tau = 1$, and $\mathrm{u}_0 = 0$, we get that the distribution p_t of X_t interpolates between $p_0 = p_{\text{data}}$ and $p_\tau = \mathrm{Poisson}(\mathrm{u}_\tau)$. Note that the choice of noise in Equation (5) guarantees that $X_t \in \mathbb{N}^D$ for all $t \in [0, \tau]$.

When p_{data} is supported over $\mathbb{Z}^D$ we perturb a data point $\mathbf{x}_{\text{data}} \sim p_{\text{data}}$ by

$$X_t := \mathbf{x}_{\text{data}} - \mathrm{L}_t + \mathrm{N}_t, \qquad \text{with} \qquad \mathrm{L}_t \sim \mathrm{Poisson}(\mathrm{d}_t), \quad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{6}$$

where $(\mathbf{x}_{\text{data}}, \mathrm{L}_t, \mathrm{N}_t)$ are independent. Taking τ to be the final time, and choosing $\mathrm{d}_0 = \mathrm{u}_0 = 0$, and $\mathrm{d}_\tau, \mathrm{u}_\tau$ large enough, we get that the distribution p_t of X_t interpolates between $p_0 = p_{\text{data}}$ and $p_\tau \overset{\text{approx.}}{\sim} \mathrm{Skellam}(\mathrm{d}_\tau, \mathrm{u}_\tau)$ (distribution of the difference of independent Poissons; Skellam, 1946). Note that the choice of noise in Equation (6) guarantees that $X_t \in \mathbb{Z}^D$ for all $t \in [0, \tau]$.

The perturbations (5) and (6) are very simple, but finding CTMCs whose marginals match these perturbations, which at the same time can take advantage of denoiser-based training, is not obvious and has been missing so far. In this work we construct CTMCs whose marginals match Equation (5) or Equation (6), which can be trained using only **latent** denoisers. Our key innovation is to train, in each of the perturbations (5) and (6), **latent denoisers** which denoise not the data itself, but **partially noised data**. We prove that training such denoisers suffices to sample from the data distribution. Further, we will show that the data denoiser can be expressed as a **non-linear** and **non-local** function of the latent denoisers. The subsequent sections show how to achieve this goal for target distributions supported on $\mathbb{N}^D$ or $\mathbb{Z}^D$.

2.1 The Binomial-Poisson process and the Poisson-Poisson process

In this section we provide our main result which shows how to construct a denoiser-based CTMC whose marginals match the perturbations in Equation (5) and Equation (6). We call the CTMC whose marginals match Equation (5) the **Binomial-Poisson process**, and the CTMC whose marginals match Equation (6) the **Poisson-Poisson process**.

In the following we let $\{e_i\}_{i \in [D]}$ be the one-hot vectors, and given continuously differentiable, coordinatewise strictly increasing noise schedules d_t (which are in $[0, 1]^D$ for (5) and in $\mathbb{R}_{\geq 0}^D$ for (6)), and $\mathrm{u}_t \in \mathbb{R}_{\geq 0}^D$, we write $\dot{\mathrm{d}}_t := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{d}_t$ and $\dot{\mathrm{u}}_t := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{u}_t$.

Theorem 2.1. *Let $(p_t)_{t \in [0,\tau]}$ be the marginals of $(X_t)_{t \in [0,\tau]}$,*

$$X_t := \mathbf{x}_{\text{data}} - \mathrm{L}_t + \mathrm{N}_t, \tag{7}$$

where

	L_t	N_t	Independence	d_0^i	u_0^i	d_τ^i
Binomial–Poisson	$\mathrm{Binomial}(\mathbf{x}_{\text{data}}, \mathrm{d}_t)$	$\mathrm{Poisson}(\mathrm{u}_t)$	$(\mathbf{x}_{\text{data}}, \mathrm{L}_t) \perp \mathrm{N}_t$	0	0	1
Poisson–Poisson	$\mathrm{Poisson}(\mathrm{d}_t)$	$\mathrm{Poisson}(\mathrm{u}_t)$	$\mathbf{x}_{\text{data}}, \mathrm{L}_t, \mathrm{N}_t$ independent	0	0	–

and define the latent denoisers

$$\mathrm{m}_{\texttt{latent-down}}(\mathbf{x}, t) := \mathbb{E}[\mathrm{L}_t | X_t = \mathbf{x}], \qquad \mathrm{m}_{\texttt{latent-up}}(\mathbf{x}, t) := \mathbb{E}[\mathrm{N}_t | X_t = \mathbf{x}]. \tag{8}$$

Write $s = \tau - t$ for reverse time. There exists a forward CTMC with the same marginals as $(X_t)_{t\in[0,\tau]}$ whose time reversal $(\mathbf{X}_s)_{s\in[0,\tau]}$ has initial distribution $\mathbf{X}_0 \sim p_\tau$ and, for $0 < s < \tau$, rate matrices

$$Q_{\mathbf{xy}}(s) = \begin{cases} \frac{\dot{\mathrm{u}}^i_{\tau-s}}{\mathrm{u}^i_{\tau-s}} \, \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, \tau - s) & \text{if} \quad \mathbf{y} = \mathbf{x} - e_i, \quad i \in [D] \\ \frac{\dot{\mathrm{d}}^i_{\tau-s}}{\mathrm{d}^i_{\tau-s}} \, \mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, \tau - s) & \text{if} \quad \mathbf{y} = \mathbf{x} + e_i, \quad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\} \\ -\sum_{\mathbf{y} \in \{\mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\}} Q_{\mathbf{xy}}(s) & \text{if} \quad \mathbf{y} = \mathbf{x} \end{cases}. \tag{9}$$

In particular,

$$\mathbf{X}_s \sim p_{\tau - s} \qquad \text{for all} \qquad s \in [0, \tau]. \tag{10}$$

The proof of Theorem 2.1 relies on combining discrete Tweedie's formulas for *both* the Binomial and Poisson distributions, and is contained in Section B-Section C.

While Equation (9) writes the rates in terms of two latent denoisers, it follows from Corollary B.5-Corollary C.4 that, at interior times, the rates can in fact be expressed using a **single** latent denoiser, since $\mathrm{m}_{\texttt{latent-up}}$ and $\mathrm{m}_{\texttt{latent-down}}$ can be expressed in terms of each other. However, this relation is **non-linear** and **non-local**, creating both numerical stability issues and more expensive neural function evaluations, which is why we opt for the formulation in Equation (9). Lastly, Corollary B.5-Corollary C.4 also show that the data denoiser $\mathrm{m}_{\texttt{data}}$ can be written as a non-linear and non-local function of a single latent denoiser.

Comparing the processes Both the Binomial-Poisson process (for distributions over $\mathbb{N}^D$) and the Poisson-Poisson process (for distributions over $\mathbb{Z}^D$) are defined in terms of `latent-up` and `latent-down` denoisers. However, in the Binomial-Poisson process the latent variable L_t depends on the data $\mathbf{x}_{\text{data}}$, while in the Poisson-Poisson process $(\mathrm{L}_t, \mathbf{x}_{\text{data}})$ are independent. This is due to the fact that the Binomial-Poisson process needs to remain in $\mathbb{N}^D$.

An additional difference is that the final time distribution of the Binomial-Poisson perturbation (5) is exactly $\mathrm{Poisson}(\mathrm{u}_\tau)$, while the final time distribution of the Poisson-Poisson perturbation (6) is only approximately $\mathrm{Skellam}(\mathrm{d}_\tau, \mathrm{u}_\tau)$.

Finally, we note that while from a theoretical perspective the Poisson-Poisson process is easier to work with, in our experiments we often find the Binomial-Poisson process to be more effective.

2.2 Training and sampling

For ease of exposition, we present the algorithms using $\tau = 1$, with $\mathrm{d}_t = t$ for Binomial-Poisson, $\mathrm{d}_t = bt$ for Poisson-Poisson, and $\mathrm{u}_t = at$ for both, where $a, b > 0$. We omit any conditioning context, which would otherwise be passed to the neural networks, as well as additional clipping to the data range.

Training We can apply two different approaches to training $\mathrm{m}_{\texttt{latent-down}}$ and $\mathrm{m}_{\texttt{latent-up}}$.

Algorithm 1. Use the fact that the optimal denoiser is, by definition, a conditional expectation so it can be trained using any valid Bregman divergence, in particular the mean-square error (MSE) loss.

Algorithm 2. For each coordinate $i \in [D]$ train a marginal posterior distribution $p_{\mathbf{x}^i_{\text{data}} | X_t}$ and then use either Lemma B.6 (Binomial-Poisson) or Lemma C.5 (Poisson-Poisson) to compute $\mathrm{m}_{\texttt{latent-down}}$ and $\mathrm{m}_{\texttt{latent-up}}$. Crucially, note that we only need to train the collection of D *marginal* posteriors $p_{\mathbf{x}^i_{\text{data}} | X_t}$, rather than the full posterior $p_{\mathbf{x}_{\text{data}} | X_t}$ which is much more complicated. This is possible thanks to Theorem 2.1 which shows that ultimately we only need the denoisers in order to sample from p_{data}.

Algorithm 1: Denoiser Training

```
# Process: Binomial-Poisson or
    Poisson-Poisson
# Noise scales a, b > 0
# L_bregman: any valid Bregman divergence
# sample_time has full support on (0, 1)
x_data = sample_data(p_data)
t = sample_time(0, 1)
L_t = Binomial(x_data, t)# Binomial-Poisson
# or L_t = Poisson(b * t) for Poisson-Poisson
N_t = Poisson(a * t)
X_t = x_data - L_t + N_t
m_L, m_N = net_latents(X_t, t)
loss = L_bregman(L_t, m_L)
     + L_bregman(N_t, m_N)
update(net_latents, loss)
```

Algorithm 2: Marginal Posterior Training

```
# Process: Binomial-Poisson or
    Poisson-Poisson
# Noise scales a, b > 0
# sample_time has full support on (0, 1)
x_data = sample_data(p_data)
t = sample_time(0, 1)
L_t = Binomial(x_data, t)# Binomial-Poisson
# or L_t = Poisson(b * t) for Poisson-Poisson
N_t = Poisson(a * t)
X_t = x_data - L_t + N_t
posterior = net_posterior(X_t, t)
for i = 1, ..., D:
    loss_i = cross_entropy(
        posterior_i, x_data^i)
loss = sum_i(loss_i)
update(net_posterior, loss)
```

Algorithm 3: Tau-leaping sampling

```
# Process: Binomial-Poisson or
    Poisson-Poisson
# Latent-mean estimator latents_denoiser;
    a, b > 0
# Number of steps T; stopping time 0 < ε < 1
h = (1 - epsilon) / T
x = Poisson(a)
if process == "Poisson-Poisson":
    x = x - Poisson(b)
for j = 0, ..., T - 1:
    t = 1 - j * h
    m_L, m_N = latents_denoiser(x, t)
    # Apply coordinatewise for every i ∈ [D]
    lambda_+^i = max(m_L^i, 0) / t
    lambda_-^i = max(m_N^i, 0) / t
    P_+^i = Poisson(h * lambda_+^i)
    P_-^i = Poisson(h * lambda_-^i)
    if process == "Binomial-Poisson":
        P_-^i = min(P_-^i, x^i)
    x^i = x^i + P_+^i - P_-^i
return x
```

Algorithm 4: Euler sampling

```
# Process: Binomial-Poisson or
    Poisson-Poisson
# Latent-mean estimator latents_denoiser;
    a, b > 0
# Number of steps T; stopping time 0 < ε < 1
h = (1 - epsilon) / T
x = Poisson(a)
if process == "Poisson-Poisson":
    x = x - Poisson(b)
for j = 0, ..., T - 1:
    t = 1 - j * h
    m_L, m_N = latents_denoiser(x, t)
    # Apply coordinatewise for every i ∈ [D]
    lambda_+^i = max(m_L^i, 0) / t
    lambda_-^i = max(m_N^i, 0) / t
    p_i = min(h * lambda_+^i, 1)
    q_i = min(h * lambda_-^i, 1)
    if process == "Binomial-Poisson":
        if x^i == 0:
            q_i = 0
    s_i = max(1, p_i + q_i)
    Delta^i = Categorical(
        +1 : p_i / s_i,
        -1 : q_i / s_i,
         0 : 1 - (p_i + q_i) / s_i)
    x^i = x^i + Delta^i
return x
```

Sampling The crux of Theorem 2.1 is that once the latent denoisers $(\mathrm{m}_{\texttt{latent-down}}, \mathrm{m}_{\texttt{latent-up}})$ are trained, we can approximately sample from p_{data} using any sampling algorithm for CTMCs, with the initial distribution $\mathrm{Poisson}(a)$ (Binomial-Poisson) or $\mathrm{Skellam}(b, a)$ (Poisson-Poisson). The Poisson initialization matches the terminal distribution exactly, whereas the Skellam initialization introduces an approximation. For example, Algorithm 3 and Algorithm 4 present the tau-leaping and Euler methods, respectively. Here, `latents_denoiser` returns estimates of $(\mathrm{m}_{\texttt{latent-down}}, \mathrm{m}_{\texttt{latent-up}})$: it calls `net_latents`, trained by regression (e.g., MSE), or converts the marginal posteriors returned by `net_posterior` using the formulas in Lemmas C.5 and B.6.

3 Experiments

We validate our method on synthetic datasets (Section 3.1), CIFAR-10 (Section 3.2), and Sudoku completion (Section 3.3). The code to reproduce all of the experiments is available at the anonymous link `https://www.github.com/AnonrsySubmitter/JUD`.

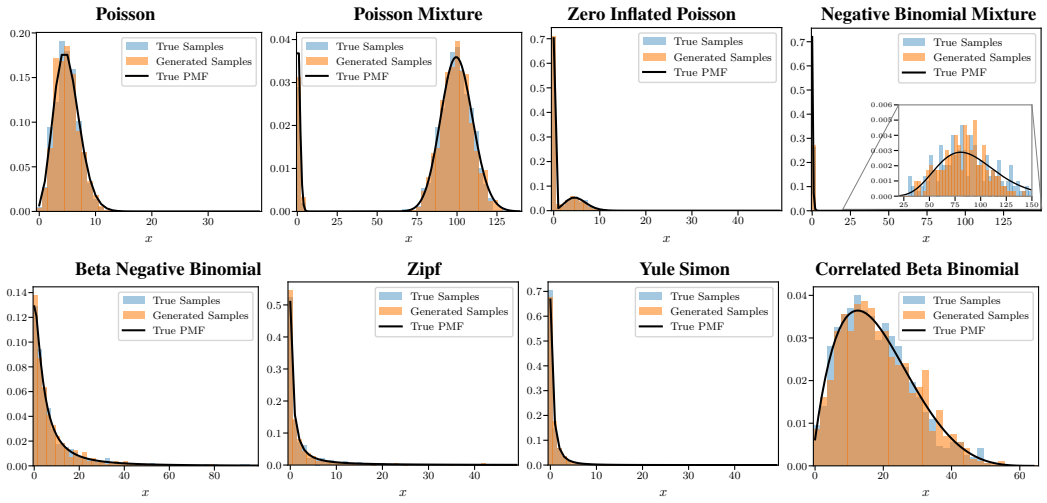

Figure 2: Target dataset and the generated samples using the Binomial-Poisson process for all eight synthetic distributions.

3.1 Synthetic Experiments

We evaluate the performance of the Binomial-Poisson and Poisson-Poisson models (with Algorithm 1) on a series of benchmark low-dimensional examples where we estimate the Wasserstein-1 distance between the true and generated distributions. For each example, we use $N = 20,000$ samples from the true distribution to learn the denoisers. At evaluation time, each trained model generates 1,000 samples via a reverse-time Euler discretization with 2,000 steps. We compute W_1 between these samples and an independent batch of 1,000 samples from the target distribution. Additional details on the data, model and training can be found in Appendix F.

Results. Table 1 presents the Wasserstein-1 distance between the generated samples and true target samples for all models. Across almost all distributions, in particular those that have multiple modes (e.g., Poisson Mixture and Negative Binomial Mixture), we observe closer agreement between each target distribution and the generated samples with our proposed Binomial-Poisson process over alternative discrete diffusion models. In Figure 2, we include a histogram and show close agreement with the Binomial-Poisson process to the tested distributions.

Table 1: Wasserstein-1 distance across distributions and sampling methods (mean ± std). Our process is denoted in bold and the best results are highlighted in orange.

	Binomial-only	Binomial-Poisson	Poisson-Poisson	Count Bridges	count-FM	SEDD
Distribution	Shenfeld et al. (2026)	**(Ours)**	**(Ours)**	Fishman et al. (2026)	Wei & Pearson (2026)	Lou et al. (2024)
Poisson	0.31 ± 0.05	**0.13 ± 0.02**	0.35 ± 0.02	0.22 ± 0.02	0.28 ± 0.06	0.46 ± 0.08
Poisson Mixture	1.52 ± 0.21	**1.39 ± 0.30**	89.71 ± 0.34	1.69 ± 0.34	1.68 ± 0.41	1.96 ± 0.33
Zero-Inflated Poisson	0.20 ± 0.03	**0.12 ± 0.01**	0.32 ± 0.02	0.16 ± 0.03	0.33 ± 0.06	0.53 ± 0.08
Negative Binomial Mixture	1.68 ± 0.23	**1.62 ± 0.56**	17.73 ± 0.41	1.80 ± 0.74	2.45 ± 0.46	2.35 ± 0.33
Beta Negative Binomial	0.87 ± 0.07	**0.65 ± 0.06**	6.14 ± 0.14	0.73 ± 0.07	1.52 ± 0.14	2.57 ± 0.14
Zipf	0.36 ± 0.02	**0.39 ± 0.03**	1.79 ± 0.10	0.64 ± 0.05	0.66 ± 0.09	1.08 ± 0.09
Yule–Simon	0.32 ± 0.04	**0.12 ± 0.01**	0.23 ± 0.02	0.45 ± 0.03	0.38 ± 0.04	0.63 ± 0.05
Correlated Beta-Binomial	0.81 ± 0.11	**0.62 ± 0.10**	11.39 ± 0.19	0.59 ± 0.09	0.63 ± 0.04	1.23 ± 0.38

3.2 CIFAR-10

We evaluate the Binomial-Poisson model of Theorem 2.1 on unconditional generation of 32×32 CIFAR-10 images (Krizhevsky et al., 2009). We use the DDPM++ architecture (Song et al., 2021b) with an EDM-style preconditioning (Karras et al., 2022), regressing the targets $(\mathrm{L}_t, \mathrm{N}_t)$; see Algorithm 1. Both predictions are produced by a single shared network with two heads. Sampling uses tau-leaping with 2048 uniformly spaced steps; see Algorithm 3. Training and sampling details are given in Sections G.1 and G.2.

We achieve a FID of 2.07 on 50,000 generated images. Table 2 compares generation quality with continuous, hybrid, and discrete diffusion models. We obtain the lowest FID among the discrete models in this comparison. Figure 3 shows the first 64 randomly sampled images from the trained model.

Table 2: Unconditional CIFAR-10 generation. FID evaluated on 50,000 generated samples.

Method	FID ↓
Continuous data models	
EDM (Karras et al., 2022)	1.97
EDM (our reproduction)	2.09
ScoreSDE (Song et al., 2021b)	2.20
IDDPM (Nichol & Dhariwal, 2021)	2.90
DDPM (Ho et al., 2020)	3.17
Hybrid data models	
CADD† (Zheng et al., 2025)	2.88
Discrete (ordered) data models	
JUD: Binomial-Poisson (ours)	**2.07**
Binomial Flow (Shenfeld et al., 2026)	2.94
Blackout (Santos et al., 2023)	4.58
LTJ (Chen & Zhou, 2023)	4.80
ItDPDM (Bhattacharya et al., 2025)	4.84
Discrete (categorical) data models	
MDM-Prime (Chao et al., 2025)	3.26
Discrete FM (Gat et al., 2024)	3.63
τ-LDR-10 (Campbell et al., 2022)	3.74
MDM (Chao et al., 2025)	4.66
MDM-Mixture (Chao et al., 2025)	4.80
D3PM (Gauss.) (Austin et al., 2021)	7.34
CTDD-DG (Nisonoff et al., 2025)	7.86

† CADD augments discrete diffusion with a paired continuous latent diffusion.

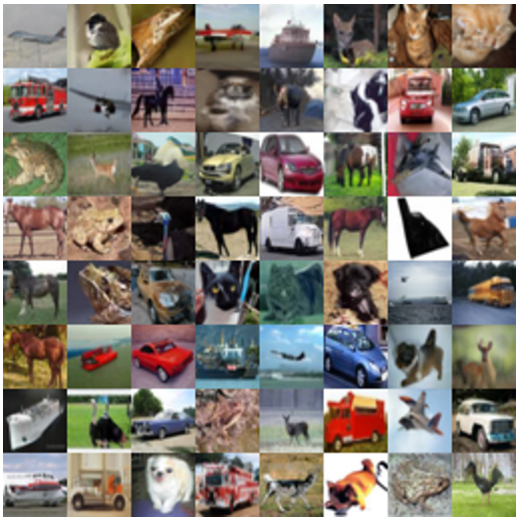

Figure 3: Unconditional CIFAR-10 samples from the Binomial-Poisson model.

Section G.3 reports FID as a function of the number of sampling steps for both samplers; at 512 steps the FID is 2.29.

3.3 SUDOKU

We test the Binomial-Poisson model of Theorem 2.1 on the hard Sudoku benchmark of Deschenaux & Gulcehre (2026). Each puzzle has 30 visible digits and a unique solution. The dataset comprises 48,000 training and 2,000 validation puzzles. The metric is exact-board accuracy: all 81 predicted digits must match the reference solution. Note that this dataset is not ordinal since Sudoku is invariant under permutations of the symbols. Our goal in this task is to explore the capabilities of our framework to general categorical data.

Model and evaluation. We use the benchmark's DiT backbone configuration. We use the same linear-time schedules as in CIFAR-10, with mean digit value 5. We train with two types of loss: **(1)** mean-square error (MSE) and **(2)** cross entropy (CE). MSE uses the same $(\mathrm{L}_t, \mathrm{N}_t)$ targets and affine preconditioning as CIFAR-10 (Tables 4 and 5); see Algorithm 1. CE instead learns a nine-class distribution for each digit, conditioned on the perturbed solution and context; see Algorithm 2. The denoiser(s) follow analytically from this posterior (Section H.2). A single network evaluation returns two scalar outputs per cell for MSE or nine logits for CE. Training configurations are given in Section H.1.

Sampling uses the tau-leaping scheme (Algorithm 3) on a uniform grid from $t = 1$ to $t = 10^{-3}$. At the final t, if training with MSE, we round and clip $X_t + \mathrm{m}_{\texttt{latent-down}}(X_t, t) - \mathrm{m}_{\texttt{latent-up}}(X_t, t)$ to

$\{1, \ldots, 9\}$, whereas if we train with CE we take the most probable digit under the posterior. We use 180 network evaluations, including the final evaluation, matching the diffusion and flow baselines.

Results. Table 3 compares our results with the published baselines. A separate block reports digit-shuffled training, which applies a random permutation of $1, \ldots, 9$ jointly to each puzzle. This preserves Sudoku constraints while varying the assignment of symbols to ordered counts. Posterior CE achieves higher mean accuracy than MSE in both settings.

Table 3: Hard Sudoku exact-board accuracy. Baselines are from Table 1 of Deschenaux & Gulcehre (2026). Our scores are means (not best) over eight independent training runs.

Method	Accuracy (%) ↑
Autoregressive models	
AR (sampled)	0.6
AR (greedy)	1.0
Discrete diffusion models	
MDLM	30.2
Duo	58.4
Continuous and hybrid models	
CANDI	16.7
CANDI (pure Gaussian)	12.5
FLM	44.5
$\mathbb{S}$-FLM	14.0
$\mathbb{S}$-FLM + $\alpha^\star(0.1)^\dagger$	43.2
$\mathbb{S}$-FLM + $\alpha^\star(0.1)$ + adaptive†	45.0
Binomial-Poisson (ours)	
MSE (noise/lost count)	**17.49**
Posterior CE	**40.31**
Binomial-Poisson (ours), *with digit shuffling*	
MSE (noise/lost count)	**29.00**
Posterior CE	**66.99**

† Schedule variants: $\alpha^\star(0.1)$ denotes truncation, with an additional adaptive noise schedule where indicated.

4 Conclusion

In this work we introduce **Jumping Up and Down (JUD)**: a **denoiser-based** family of generative diffusion models for discrete ordinal data (either $\mathbb{N}^D$ or $\mathbb{Z}^D$) which allows for **bi-directional perturbations**. Our main result (Theorem 2.1) shows that it suffices to learn **latent denoisers** in order to sample from the data distribution. Further, we show that theoretically a **single** denoiser suffices to sample from the data distribution, as the two latent denoisers can be expressed in terms of each other. In addition, we also prove that the **data denoiser** can be expressed in terms of the latent denoiser(s). We validate our methods on synthetic datasets, CIFAR-10, and the Sudoku completion task. In particular, on synthetic datasets we often obtain superior results over alternative discrete diffusion models, and we achieve a state-of-the-art result among discrete diffusion models for CIFAR-10.

5 Limitations

(1) For the purpose of generative modeling considered in this work, the source distribution should be easy to sample (which is the case for us, as it is simply a Poisson or approximately Skellam). But in some applications (e.g., Fishman et al. (2026)) we would like to have an arbitrary source measure, which our method does not allow for. (2) Currently, our method requires a high number of neural function evaluations (NFE) for sampling. (3) Compared to the Binomial-Poisson method, the Poisson-Poisson method has an extra parameter which needs to be tuned, namely the final time τ, in order to make the approximation $q_\tau \overset{\text{approx.}}{\sim} \mathrm{Skellam}(\mathrm{d}_\tau, \mathrm{u}_\tau)$ more accurate.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs, specifically to carry out or verify calculations in individual proof steps (such as the EDM scalings in Section E), and to implement methods, specifically to help write the code for the experiments. We have not used generative AI tools to generate synthetic datasets, to develop the methodology or the theoretical framework, to formulate mathematical claims, to devise proof strategies, to propose or refine hypotheses, to design the experiments, or to interpret results; translation, dataset cleaning and reformatting, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the paper for readability. We have reviewed all AI-assisted work: every AI-assisted calculation was checked by the authors, and all AI-assisted code was reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work is methodological in nature and uses public datasets. We declare no ethical issues corresponding to human subjects, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues. While generative AI tools can cause harm, our work in this context is at the research scale.

REPRODUCIBILITY STATEMENT

We provide extensive details on our experiments in the paper itself and the appendices. We also provide an anonymized link to the code producing the experiments: `https://www.github.com/AnonrsySubmitter/JUD`.

REFERENCES

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. URL `https://openreview.net/pdf?id=h7-XixPCAL`.

Francis Bach and Saeed Saremi. Sampling binary data by denoising through score functions. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=4ESwnU0kCA`.

Gabriel Bénédict, Olivier Jeunen, Samuele Papa, Samarth Bhargav, Daan Odijk, and Maarten de Rijke. RecFusion: A binomial diffusion process for 1D data for recommendation. *arXiv preprint arXiv:2306.08947*, 2023.

Sagnik Bhattacharya, Abhiram Gorle, Ahsan Bilal, Connor Ding, Amit Kumar Singh Yadav, and Tsachy Weissman. ItDPDM: Information-theoretic discrete Poisson diffusion model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=6ZkcC9NmGU`.

Sanjukta Bhattacharya, Christian Gensbigler, Shaamil Karim, and Jon Lees. Discrete diffusion for single-cell gene expression modeling. *bioRxiv*, 2026. doi: 10.64898/2026.02.19.705033.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=DmT862YAieY`.

Andrew Campbell, William Harvey, Christian Weilbach, Valentin De Bortoli, Tom Rainforth, and Arnaud Doucet. Trans-dimensional generative modeling via jump diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=t6nA7x3GAC`.

Chen-Hao Chao, Wei-Fang Sun, Hanwen Liang, Chun-Yi Lee, and Rahul G. Krishnan. Beyond masked and unmasked: Discrete diffusion models via partial masking. *arXiv preprint arXiv:2505.18495*, 2025. doi: 10.48550/arXiv.2505.18495. URL https://arxiv.org/abs/2505.18495. Published at NeurIPS 2025.

Tianqi Chen and Mingyuan Zhou. Learning to jump: Thinning and thickening latent counts for generative modeling. In *International Conference on Machine Learning*, pp. 5367–5382. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23ap/chen23ap.pdf.

Chaoran Cheng, Zhanghan Ni, Yanru Qu, Yuxin Chen, Ruihan Guo, Jiajun Fan, and Ge Liu. Variable-length generative protein design via generalized Poisson flow. *arXiv preprint arXiv:2607.09039*, 2026.

Justin Deschenaux and Caglar Gulcehre. Language modeling with hyperspherical flows. *arXiv preprint arXiv:2605.11125*, 2026. URL https://arxiv.org/abs/2605.11125.

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

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

Florentin Guth, Zahra Kadkhodaie, and Eero Simoncelli. Learning normalized image densities via dual score matching. *Advances in Neural Information Processing Systems*, 38:89796–89826, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

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

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=6nbpPqUCIi7.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. ELF: Embedded language flows. *arXiv preprint arXiv:2605.10938*, 2026.

Zahra Kadkhodaie, Aram-Alexandre Pooladian, Sinho Chewi, and Eero Simoncelli. Blind denoising diffusion models and the blessings of dimensionality. *arXiv preprint arXiv:2602.09639*, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. URL https://openreview.net/pdf?id=k7FuTOWMOc7.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 and CIFAR-100 datasets. http://www.cs.toronto.edu/~kriz/cifar.html, 2009.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

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

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_RL7wtHkPJK.

Andrea Montanari. Sampling, diffusions, and stochastic localization. *arXiv preprint arXiv:2305.10690*, 2023. URL https://arxiv.org/abs/2305.10690.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a/nichol21a.pdf.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2406.01572. URL https://openreview.net/pdf?id=XsgHl54yO7. arXiv:2406.01572.

Le-Tuyet-Nhi Pham, Dario Shariatian, Antonio Ocello, Giovanni Conforti, and Alain Oliviero Durmus. Discrete Markov probabilistic models: An improved discrete score-based framework with sharp convergence bounds under minimal assumptions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=biJiSMLGOV.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *Advances in Neural Information Processing Systems*, 37: 36046–36070, 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.

Javier E. Santos, Agnese Marcato, Roman Colman, Nicholas Lubbers, and Yen Ting Lin. Discrete spatial diffusion: Intensity-preserving diffusion modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kz3w2A2y0e.

Yair Shenfeld, Ricardo Baptista, and Stefano Peluchetti. Binomial flows: Denoising and flow matching for discrete ordinal data. *arXiv preprint arXiv:2605.00360*, 2026.

John G Skellam. The frequency distribution of the difference between two Poisson variates belonging to different populations. *Journal of the Royal Statistical Society*, 109(3):296, 1946.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in neural information processing systems*, volume 32, 2019.

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

Qiao Sun, Zhicheng Jiang, Hanhong Zhao, and Kaiming He. Is noise conditioning necessary for denoising generative models? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=pTSWi6RTtJ.

Chloe Wang, Mehran Karimzadeh, Neal G Ravindra, Lexi R Bounds, Nader Alerasool, Ann C Huang, Shihao Ma, Daniel R Gulbranson, Haotian Cui, Yongju Lee, et al. X-Cell: Scaling causal perturbation prediction across diverse cellular contexts via diffusion language models. *bioRxiv*, 2026a. doi: 10.64898/2026.03.18.712807.

Mingxuan Wang, Cheng Chen, Gaoyang Jiang, Zijia Ren, Chuangxin Zhao, Lu Shi, and Yanbiao Ma. ScDiVa: Masked discrete diffusion for joint modeling of single-cell identity and expression. *arXiv preprint arXiv:2602.03477*, 2026b.

Runqian Wang and Yilun Du. Equilibrium matching: Generative modeling with implicit energy-based models. *arXiv preprint arXiv:2510.02300*, 2025.

Ganchao Wei and John Pearson. Flow matching for count data. *arXiv preprint arXiv:2605.07746*, 2026.

Ludwig Winkler, Lorenz Richter, and Manfred Opper. Bridging discrete and continuous state spaces: Exploring the Ehrenfest process in time-continuous diffusion models. *arXiv preprint arXiv:2405.03549*, 2024.

Han Zhang, Guo-Hua Yuan, Chaohao Yuan, Tingyang Xu, Tian Bian, Hong Cheng, Wenbing Huang, Deli Zhao, and Yu Rong. Lingshu-Cell: A generative cellular world model for transcriptome modeling toward virtual cells. *arXiv preprint arXiv:2603.25240*, 2026.

Huangjie Zheng, Shansan Gong, Ruixiang Zhang, Tianrong Chen, Jiatao Gu, Mingyuan Zhou, Navdeep Jaitly, and Yizhe Zhang. Continuously augmented discrete diffusion model for categorical generative modeling. *arXiv preprint arXiv:2510.01329*, 2025. doi: 10.48550/arXiv.2510.01329. URL https://arxiv.org/abs/2510.01329.

A Poisson and Binomial distributions

In this section we derive some properties of the Poisson and Binomial distributions which we will repeatedly use. To ease the notation we set $[D] := \{1, \dots, D\}$ and recall that $\{e_i\}_{i=1}^D$ is the standard basis. We denote

$$\Pi_\alpha := \mathrm{Poisson}(\alpha) \qquad \text{and} \qquad \mathcal{B}_{\mathbf{n},r} := \mathrm{Binomial}(\mathbf{n}; r), \tag{11}$$

for $\alpha \in \mathbb{R}^D_{\geq 0}$, $\mathbf{n} \in \mathbb{N}^D$, and $r \in [0,1]^D$, i.e.,

$$\Pi_\alpha(\mathbf{x}) = \prod_{i=1}^D \Pi_{\alpha^i}(\mathbf{x}^i) = \prod_{i=1}^D e^{-\alpha^i} \frac{(\alpha^i)^{\mathbf{x}^i}}{\mathbf{x}^i!}, \tag{12}$$

and

$$\mathcal{B}_{\mathbf{n},r}(\mathbf{x}) = \prod_{i=1}^D \mathcal{B}_{\mathbf{n}^i,r^i}(\mathbf{x}^i) = \prod_{i=1}^D 1_{\mathbf{x}^i \leq \mathbf{n}^i} \binom{\mathbf{n}^i}{\mathbf{x}^i} (r^i)^{\mathbf{x}^i} (1-r^i)^{\mathbf{n}^i - \mathbf{x}^i}. \tag{13}$$

We will use the convention $\Pi_\alpha(\mathbf{x}) = 0$ for $\mathbf{x} \notin \mathbb{N}^D$ and $\mathcal{B}_{\mathbf{n},r}(\mathbf{x}) = 0$ if $x^i \notin \{0, \dots, n^i\}$ for some $i \in [D]$.

Lemma A.1 (Properties of Poisson distribution)**.** *Let $[0,\infty) \ni t \mapsto \alpha_t \in \mathbb{R}^D_{\geq 0}$ be differentiable. Then, at times when $\alpha_t \in \mathbb{R}^D_{>0}$,*

$$\partial_t \Pi_{\alpha_t}(\mathbf{x}) = \Pi_{\alpha_t}(\mathbf{x}) \sum_{i=1}^D \left(\mathbf{x}^i \frac{\dot\alpha_t^i}{\alpha_t^i} - \dot\alpha_t^i \right), \qquad \mathbf{x} \in \mathbb{N}^D, \tag{14}$$

and

$$\Pi_{\alpha_t}(\mathbf{x} - e_i) = \frac{\mathbf{x}^i}{\alpha_t^i} \Pi_{\alpha_t}(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{N}^D, \qquad i \in [D]. \tag{15}$$

Moreover, for all $t \geq 0$,

$$\partial_t \Pi_{\alpha_t}(\mathbf{x}) = \sum_{i=1}^{D} \dot{\alpha}_t^i \left(\Pi_{\alpha_t}(\mathbf{x} - e_i) - \Pi_{\alpha_t}(\mathbf{x})\right), \qquad \mathbf{x} \in \mathbb{N}^D. \tag{16}$$

Proof. Equation (14) follows from

$$\partial_t \log \Pi_{\alpha_t}(\mathbf{x}) = \partial_t \sum_{i=1}^{D} \log\left(e^{-\alpha_t^i} \frac{(\alpha_t^i)^{\mathbf{x}^i}}{\mathbf{x}^i!}\right) = \sum_{i=1}^{D} \left(\mathbf{x}^i \frac{\dot{\alpha}_t^i}{\alpha_t^i} - \dot{\alpha}_t^i\right),$$

while Equation (15) follows from

$$\Pi_{\alpha_t}(\mathbf{x} - e_i) = e^{-\alpha_t^i} \frac{(\alpha_t^i)^{(\mathbf{x}^i - 1)}}{(\mathbf{x}^i - 1)!} \prod_{j \neq i} \Pi_{\alpha_t^j}(\mathbf{x}^j) = \frac{\mathbf{x}^i}{\alpha_t^i} \Pi_{\alpha_t^i}(\mathbf{x}^i) \prod_{j \neq i} \Pi_{\alpha_t^j}(\mathbf{x}^j).$$

Equation (16) follows from the combination of (14) and (15). At zero parameter values, it follows by direct differentiation of the probability mass. □

Lemma A.2 (Properties of Binomial distribution)**.** *Fix $\mathbf{n} \in \mathbb{N}^D$ and let $[0, \infty) \ni t \mapsto r_t \in [0, 1]^D$ be differentiable. Then, at times when $r_t \in (0, 1)^D$,*

$$\partial_t \mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x}) = \mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x}) \sum_{i=1}^{D} \left(\mathbf{x}^i \frac{\dot{r}_t^i}{r_t^i} - (\mathbf{n}^i - \mathbf{x}^i) \frac{\dot{r}_t^i}{1 - r_t^i}\right), \qquad \mathbf{x} \in \mathbb{N}^D, \tag{17}$$

and

$$\mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x} + e_i) = \frac{\mathbf{n}^i - \mathbf{x}^i}{\mathbf{x}^i + 1} \frac{r_t^i}{1 - r_t^i} \mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{N}^D, \qquad i \in [D]. \tag{18}$$

Consequently,

$$\partial_t \mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x}) = \sum_{i=1}^{D} \frac{\dot{r}_t^i}{r_t^i} \left(\mathbf{x}^i \mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x}) - (\mathbf{x}^i + 1) \mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x} + e_i)\right), \qquad \mathbf{x} \in \mathbb{N}^D. \tag{19}$$

Proof. Equation (17) follows from

$$\begin{aligned}
\partial_t \log \mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x}) &= \partial_t \sum_{i=1}^{D} \log\left(1_{\mathbf{x}^i \leq \mathbf{n}^i} \binom{\mathbf{n}^i}{\mathbf{x}^i} (r_t^i)^{\mathbf{x}^i} (1 - r_t^i)^{\mathbf{n}^i - \mathbf{x}^i}\right) \\
&= \partial_t \sum_{i=1}^{D} \left[\mathbf{x}^i \log(r_t^i) + (\mathbf{n}^i - \mathbf{x}^i) \log(1 - r_t^i)\right] = \sum_{i=1}^{D} \left(\mathbf{x}^i \frac{\dot{r}_t^i}{r_t^i} - (\mathbf{n}^i - \mathbf{x}^i) \frac{\dot{r}_t^i}{1 - r_t^i}\right),
\end{aligned}$$

while Equation (18) follows from

$$\begin{aligned}
\mathcal{B}_{\mathbf{n}, r_t}(\mathbf{x} + e_i) &= 1_{\mathbf{x}^i + 1 \leq \mathbf{n}^i} \binom{\mathbf{n}^i}{\mathbf{x}^i + 1} (r_t^i)^{\mathbf{x}^i + 1} (1 - r_t^i)^{\mathbf{n}^i - \mathbf{x}^i - 1} \prod_{j \neq i} \mathcal{B}_{\mathbf{n}^j, r_t^j}(\mathbf{x}^j) \\
&= 1_{\mathbf{x}^i + 1 \leq \mathbf{n}^i} \frac{\mathbf{n}^i - \mathbf{x}^i}{\mathbf{x}^i + 1} \frac{r_t^i}{1 - r_t^i} \mathcal{B}_{\mathbf{n}^i, r_t^i}(\mathbf{x}^i) \prod_{j \neq i}^{D} \mathcal{B}_{\mathbf{n}^j, r_t^j}(\mathbf{x}^j).
\end{aligned}$$

Equation (19) follows from the combination of (17) and (18). □

Lemma A.3 (Convolution of Binomial and Poisson)**.** *Fix $\mathbf{n} \in \mathbb{N}^D$ and let $[0, \infty) \ni t \mapsto r_t \in [0, 1]^D$ and $[0, \infty) \ni t \mapsto \alpha_t \in \mathbb{R}^D_{\geq 0}$ be differentiable. Then, for $\mathbf{x} \in \mathbb{N}^D$ and times when $r_t \in (0, 1)^D$ and $\alpha_t \in \mathbb{R}^D_{>0}$,*

$$\begin{aligned}
&\partial_t (\mathcal{B}_{\mathbf{n}, r_t} * \Pi_{\alpha_t})(\mathbf{x}) \\
&= \sum_{i=1}^{D} \left(\frac{\dot{r}_t^i}{r_t^i (1 - r_t^i)} \mathbf{x}^i - \dot{\alpha}_t^i - \frac{\dot{r}_t^i}{1 - r_t^i} \mathbf{n}^i\right) (\mathcal{B}_{\mathbf{n}, r_t} * \Pi_{\alpha_t})(\mathbf{x}) \\
&- \sum_{i=1}^{D} \left(\frac{\dot{r}_t^i}{r_t^i (1 - r_t^i)} - \frac{\dot{\alpha}_t^i}{\alpha_t^i}\right) \alpha_t^i \, (\mathcal{B}_{\mathbf{n}, r_t} * \Pi_{\alpha_t})(\mathbf{x} - e_i).
\end{aligned} \tag{20}$$

Proof. We start with a one-dimensional convolution. For fixed $i \in [D]$,

$$\begin{aligned}
\partial_t(\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i) &= \partial_t \sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} \mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i)\,\Pi_{\alpha_t^i}(\mathbf{x}^i-\mathbf{k}^i) \\
&= \sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} \left\{ \Pi_{\alpha_t^i}(\mathbf{x}^i-\mathbf{k}^i)\,\mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i) \left(\mathbf{k}^i \frac{\dot r_t^i}{r_t^i} - (\mathbf{n}^i-\mathbf{k}^i)\frac{\dot r_t^i}{1-r_t^i} \right) \right\} \\
&+ \sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} \left\{ \Pi_{\alpha_t^i}(\mathbf{x}^i-\mathbf{k}^i)\,\mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i) \left((\mathbf{x}^i-\mathbf{k}^i)\frac{\dot\alpha_t^i}{\alpha_t^i} - \dot\alpha_t^i \right) \right\} \\
&= \left(\frac{\dot\alpha_t^i}{\alpha_t^i}\mathbf{x}^i - \dot\alpha_t^i - \mathbf{n}^i \frac{\dot r_t^i}{1-r_t^i} \right) (\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i) \\
&+ \left(\frac{\dot r_t^i}{r_t^i(1-r_t^i)} - \frac{\dot\alpha_t^i}{\alpha_t^i} \right) \sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} \mathbf{k}^i\,\mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i)\,\Pi_{\alpha_t^i}(\mathbf{x}^i-\mathbf{k}^i).
\end{aligned}$$

By Equation (15),

$$\begin{aligned}
&\sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} \mathbf{k}^i\,\mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i)\,\Pi_{\alpha_t^i}(\mathbf{x}^i-\mathbf{k}^i) \\
&= \mathbf{x}^i \sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} \mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i)\,\Pi_{\alpha_t^i}(\mathbf{x}^i-\mathbf{k}^i) - \sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} (\mathbf{x}^i-\mathbf{k}^i)\,\Pi_{\alpha_t^i}(\mathbf{x}^i-\mathbf{k}^i)\mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i) \\
&= \mathbf{x}^i(\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i) - \alpha_t^i \sum_{\mathbf{k}^i=0}^{\mathbf{n}^i} \Pi_{\alpha_t^i}(\mathbf{x}^i-1-\mathbf{k}^i)\,\mathcal{B}_{\mathbf{n}^i,r_t^i}(\mathbf{k}^i) \\
&= \mathbf{x}^i(\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i) - \alpha_t^i\,(\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i-1),
\end{aligned}$$

so we conclude

$$\begin{aligned}
&\partial_t(\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i) = \\
&\left(\frac{\dot r_t^i}{r_t^i(1-r_t^i)}\mathbf{x}^i - \dot\alpha_t^i - \frac{\dot r_t^i}{1-r_t^i}\mathbf{n}^i \right) (\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i) - \left(\frac{\dot r_t^i}{r_t^i(1-r_t^i)} - \frac{\dot\alpha_t^i}{\alpha_t^i} \right) \alpha_t^i\,(\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i-1).
\end{aligned}$$

It follows that

$$\begin{aligned}
\partial_t(\mathcal{B}_{\mathbf{n},r_t} * \Pi_{\alpha_t})(\mathbf{x}) &= \sum_{i=1}^{D} \left(\prod_{j\neq i}^{D} (\mathcal{B}_{\mathbf{n}^j,r_t^j} * \Pi_{\alpha_t^j})(\mathbf{x}^j) \right) \partial_t(\mathcal{B}_{\mathbf{n}^i,r_t^i} * \Pi_{\alpha_t^i})(\mathbf{x}^i) \\
&= \sum_{i=1}^{D} \left(\frac{\dot r_t^i}{r_t^i(1-r_t^i)}\mathbf{x}^i - \dot\alpha_t^i - \frac{\dot r_t^i}{1-r_t^i}\mathbf{n}^i \right) (\mathcal{B}_{\mathbf{n},r_t} * \Pi_{\alpha_t})(\mathbf{x}) \\
&- \sum_{i=1}^{D} \left(\frac{\dot r_t^i}{r_t^i(1-r_t^i)} - \frac{\dot\alpha_t^i}{\alpha_t^i} \right) \alpha_t^i\,(\mathcal{B}_{\mathbf{n},r_t} * \Pi_{\alpha_t})(\mathbf{x}-e_i).
\end{aligned}$$

□

B THE BINOMIAL-POISSON PROCESS

The main result of this section is the Binomial-Poisson part of Theorem 2.1:

Theorem B.1 (The Binomial-Poisson process)**.** *Under the schedule assumptions of Theorem 2.1, set the noise schedules* $\mathrm{d}_t \in [0,1]^D, \mathrm{u}_t \in \mathbb{R}^D_{\geq 0}$ *to satisfy*

$$\mathrm{d}_0^i = 0, \quad \mathrm{d}_\tau^i = 1, \quad \mathrm{u}_0^i = 0, \qquad \dot{\mathrm{u}}_t^i := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{u}_t^i, \quad \dot{\mathrm{d}}_t^i := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{d}_t^i, \qquad i \in [D]. \tag{21}$$

Let $(p_t)_{t\in[0,\tau]}$ *be the marginals of* $(X_t)_{t\in[0,\tau]}$*,*

$$X_t := \mathbf{x}_{\text{data}} - \mathrm{L}_t + \mathrm{N}_t, \qquad \text{with} \qquad \mathrm{L}_t \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}, \mathrm{d}_t), \quad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{22}$$

where N_t *is independent of* $(\mathbf{x}_{\text{data}}, \mathrm{L}_t)$*, and define the latent denoisers*

$$\mathrm{m}_{\texttt{latent-down}}(\mathbf{x}, t) := \mathbb{E}[\mathrm{L}_t | X_t = \mathbf{x}], \qquad \mathrm{m}_{\texttt{latent-up}}(\mathbf{x}, t) := \mathbb{E}[\mathrm{N}_t | X_t = \mathbf{x}]. \tag{23}$$

Write $s = \tau - t$ *for reverse time. The time reversal* $(\mathbf{X}_s)_{s\in[0,\tau]}$ *of the forward CTMC in Proposition B.8 has initial distribution* $\mathbf{X}_0 \sim p_\tau = \mathrm{Poisson}(\mathrm{u}_\tau)$ *and, for* $0 < s < \tau$*, rate matrices*

$$Q_{\mathbf{xy}}(s) = \begin{cases} \frac{\dot{\mathrm{u}}^i_{\tau-s}}{\mathrm{u}^i_{\tau-s}} \, \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, \tau - s) & \text{if} \quad \mathbf{y} = \mathbf{x} - e_i, \qquad i \in [D] \\ \frac{\dot{\mathrm{d}}^i_{\tau-s}}{\mathrm{d}^i_{\tau-s}} \, \mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, \tau - s) & \text{if} \quad \mathbf{y} = \mathbf{x} + e_i, \qquad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\} \\ -\sum_{\mathbf{y} \in \{\mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\}} Q_{\mathbf{xy}}(s) & \text{if} \quad \mathbf{y} = \mathbf{x} \end{cases}. \tag{24}$$

In particular,

$$\mathbf{X}_s \sim p_{\tau - s} \qquad \text{for all} \qquad s \in [0, \tau]. \tag{25}$$

We start by simplifying the notation and write

$$\mathrm{B}_t := \mathbf{x}_{\text{data}} - \mathrm{L}_t \qquad \text{and} \qquad l_t := 1 - \mathrm{d}_t, \tag{26}$$

so in this section we focus on the perturbations of $\mathbf{x}_{\text{data}} \sim p_{\text{data}}$, for a given $t \in [0, \tau]$, by

$$X_t := \mathrm{B}_t + \mathrm{N}_t, \qquad \text{with} \qquad \mathrm{B}_t \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}, l_t), \quad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{27}$$

where N_t is independent of $(\mathbf{x}_{\text{data}}, \mathrm{B}_t)$. We will use

$$\mathrm{m}_{\texttt{latent}}(\mathbf{x}, t) := \mathbb{E}[\mathrm{B}_t | X_t = \mathbf{x}] = \mathbf{x} - \mathbb{E}[\mathrm{N}_t | X_t = \mathbf{x}], \tag{28}$$

and note that we have the relations

$$\mathrm{m}_{\texttt{latent-down}}(\mathbf{x}, t) = \mathrm{m}_{\texttt{data}}(\mathbf{x}, t) - \mathrm{m}_{\texttt{latent}}(\mathbf{x}, t), \tag{29}$$

$$\mathrm{m}_{\texttt{latent-up}}(\mathbf{x}, t) = \mathbf{x} - \mathrm{m}_{\texttt{latent}}(\mathbf{x}, t). \tag{30}$$

We begin by recording some of the conditional distributions of $(\mathbf{x}_{\text{data}}, \mathrm{B}_t, \mathrm{N}_t)$.

Lemma B.2. *Let* p_t *be the distribution of* X_t *given in* (27)*. For* $\mathbf{x} \in \mathbb{N}^D$*,*

$$p_t(\mathbf{x}) = \sum_{\mathbf{x}_0 \in \mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)(\mathcal{B}_{\mathbf{x}_0, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x}). \tag{31}$$

Fix $\mathbf{x} \in \mathbb{N}^D$*, then, for* $\mathbf{y} \in \mathbb{N}^D$*,*

$$p_{\mathrm{B}_t | X_t = \mathbf{x}}(\mathbf{y}) = \frac{\Pi_{\mathrm{u}_t}(\mathbf{x} - \mathbf{y})}{p_t(\mathbf{x})} \sum_{\mathbf{x}_0 \in \mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \mathcal{B}_{\mathbf{x}_0, l_t}(\mathbf{y}). \tag{32}$$

Fix $\mathbf{x}, \mathbf{z} \in \mathbb{N}^D$*, then, for* $\mathbf{y} \in \mathbb{N}^D$*,*

$$p_{\mathrm{B}_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}) = \frac{\mathcal{B}_{\mathbf{z}, l_t}(\mathbf{y}) \Pi_{\mathrm{u}_t}(\mathbf{x} - \mathbf{y})}{(\mathcal{B}_{\mathbf{z}, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x})}, \tag{33}$$

$$p_{\mathrm{L}_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}) = \frac{\mathcal{B}_{\mathbf{z}, \mathrm{d}_t}(\mathbf{y}) \Pi_{\mathrm{u}_t}(\mathbf{x} + \mathbf{y} - \mathbf{z})}{\sum_{\tilde{\mathbf{y}} \in \mathbb{N}^D} \mathcal{B}_{\mathbf{z}, \mathrm{d}_t}(\tilde{\mathbf{y}}) \Pi_{\mathrm{u}_t}(\mathbf{x} + \tilde{\mathbf{y}} - \mathbf{z})}, \tag{34}$$

$$p_{\mathrm{N}_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}) = \frac{\mathcal{B}_{\mathbf{z}, \mathrm{d}_t}(\mathbf{z} + \mathbf{y} - \mathbf{x}) \Pi_{\mathrm{u}_t}(\mathbf{y})}{\sum_{\tilde{\mathbf{y}} \in \mathbb{N}^D} \mathcal{B}_{\mathbf{z}, \mathrm{d}_t}(\mathbf{z} + \tilde{\mathbf{y}} - \mathbf{x}) \Pi_{\mathrm{u}_t}(\tilde{\mathbf{y}})}. \tag{35}$$

Fix $\mathbf{y} \in \mathbb{N}^D$*, then, for* $\mathbf{x} \in \mathbb{N}^D$*,*

$$p_{\mathbf{x}_{\text{data}} | X_t = \mathbf{y}}(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_t(\mathbf{y})} (\mathcal{B}_{\mathbf{x}, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{y}). \tag{36}$$

Proof. Equation (31) holds by the definition (27). For Equation (32), since $X_t|\mathrm{B}_t \sim \Pi_{\mathrm{u}_t}(\cdot - \mathrm{B}_t)$, Bayes' rule gives

$$p_{\mathrm{B}_t|X_t=\mathbf{x}}(\mathbf{y}) = \frac{p_{\mathrm{B}_t}(\mathbf{y})}{p_{X_t}(\mathbf{x})} p_{X_t|\mathrm{B}_t=\mathbf{y}}(\mathbf{x}) = \frac{\sum_{\mathbf{x}_0 \in \mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})}{p_t(\mathbf{x})} \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y}).$$

For Equation (33) we use Bayes' rule and the independence of $(\mathbf{x}_{\text{data}}, \mathrm{B}_t)$ and N_t to write

$$\begin{aligned} p_{\mathrm{B}_t|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}) &= \frac{p_{\mathrm{B}_t,X_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{x},\mathbf{z})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} = \frac{p_{\mathrm{B}_t,\mathrm{N}_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{x}-\mathbf{y},\mathbf{z})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} \\ &= \frac{p_{\mathrm{B}_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{z}) p_{\mathrm{N}_t}(\mathbf{x}-\mathbf{y})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} = \frac{p_{\text{data}}(\mathbf{z}) p_{\mathrm{B}_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}) p_{\mathrm{N}_t}(\mathbf{x}-\mathbf{y})}{p_{\text{data}}(\mathbf{z}) p_{X_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{x})} \\ &= \frac{\mathcal{B}_{\mathbf{z},l_t}(\mathbf{y})\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})}{(\mathcal{B}_{\mathbf{z},l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x})}. \end{aligned}$$

Similarly, for Equation (34) and Equation (35) we have

$$\begin{aligned} p_{\mathrm{L}_t|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}) &= \frac{p_{\mathrm{L}_t,X_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{x},\mathbf{z})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} = \frac{p_{\mathrm{L}_t,\mathrm{N}_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{x}+\mathbf{y}-\mathbf{z},\mathbf{z})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} \\ &= \frac{p_{\mathrm{L}_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{z}) p_{\mathrm{N}_t}(\mathbf{x}+\mathbf{y}-\mathbf{z})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} = \frac{p_{\text{data}}(\mathbf{z})\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\mathbf{y})\Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{z})}{p_{\text{data}}(\mathbf{z}) p_{X_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{x})} \\ &= \frac{\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\mathbf{y})\Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{z})}{p_{X_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{x})} = \frac{\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\mathbf{y})\Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{z})}{\sum_{\tilde{\mathbf{y}}\in\mathbb{N}^D}\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\tilde{\mathbf{y}})\Pi_{\mathrm{u}_t}(\mathbf{x}+\tilde{\mathbf{y}}-\mathbf{z})}, \end{aligned}$$

and

$$\begin{aligned} p_{\mathrm{N}_t|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}) &= \frac{p_{\mathrm{N}_t,X_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{x},\mathbf{z})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} = \frac{p_{\mathrm{N}_t,\mathrm{L}_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{z}+\mathbf{y}-\mathbf{x},\mathbf{z})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} \\ &= \frac{p_{\mathrm{L}_t,\mathbf{x}_{\text{data}}}(\mathbf{z}+\mathbf{y}-\mathbf{x},\mathbf{z}) p_{\mathrm{N}_t}(\mathbf{y})}{p_{X_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} = \frac{p_{\text{data}}(\mathbf{z})\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\mathbf{z}+\mathbf{y}-\mathbf{x})\Pi_{\mathrm{u}_t}(\mathbf{y})}{p_{\text{data}}(\mathbf{z}) p_{X_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{x})} \\ &= \frac{\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\mathbf{z}+\mathbf{y}-\mathbf{x})\Pi_{\mathrm{u}_t}(\mathbf{y})}{p_{X_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{x})} = \frac{\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\mathbf{z}+\mathbf{y}-\mathbf{x})\Pi_{\mathrm{u}_t}(\mathbf{y})}{\sum_{\tilde{\mathbf{y}}\in\mathbb{N}^D}\mathcal{B}_{\mathbf{z},\mathrm{d}_t}(\mathbf{z}+\tilde{\mathbf{y}}-\mathbf{x})\Pi_{\mathrm{u}_t}(\tilde{\mathbf{y}})}. \end{aligned}$$

For Equation (36) we also use Bayes' rule,

$$p_{\mathbf{x}_{\text{data}}|X_t=\mathbf{y}}(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{X_t}(\mathbf{y})} p_{X_t|\mathbf{x}_{\text{data}}=\mathbf{x}}(\mathbf{y}) = \frac{p_{\text{data}}(\mathbf{x})}{p_t(\mathbf{y})} (\mathcal{B}_{\mathbf{x},l_t} * \Pi_{\mathrm{u}_t})(\mathbf{y}).$$

□

B.1 Discrete Tweedie's formula

In this section we derive a discrete Tweedie's formula for the Binomial-Poisson perturbation (27). This formula is crucial in proving Theorem B.1.

Lemma B.3 (Discrete Tweedie's formula)**.** *Fix $t \in (0,\tau)$, and let $X_t := \mathrm{B}_t + \mathrm{N}_t$ where*

$$\mathbf{x}_{\text{data}} \sim p_{\text{data}}, \quad \mathrm{B}_t|\mathbf{x}_{\text{data}} \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}, l_t), \quad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{37}$$

with N_t independent of $(\mathbf{x}_{\text{data}}, \mathrm{B}_t)$. Denote by p_t the distribution of X_t, i.e., $X_t \sim p_t$, and define

$$\mathrm{m}_{\texttt{latent}}(\mathbf{x},t) := \mathbb{E}[\mathrm{B}_t|X_t=\mathbf{x}] = \mathbf{x} - \mathbb{E}[\mathrm{N}_t|X_t=\mathbf{x}]. \tag{38}$$

Then,

$$\frac{p_t(\mathbf{x}-e_i)}{p_t(\mathbf{x})} = \frac{\mathbf{x}^i - \mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t)}{\mathrm{u}^i_t}, \qquad \mathbf{x}\in\mathbb{N}^D, \quad t\in(0,\tau), \quad i\in[D]. \tag{39}$$

Proof.

$$\mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t) = \sum_{\mathbf{y}\in\mathbb{N}^D} \mathbf{y}^i p_{\mathrm{B}_t|X_t=\mathbf{x}}(\mathbf{y}) \overset{32}{=} \frac{1}{p_t(\mathbf{x})} \sum_{\mathbf{x}_0\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \sum_{\mathbf{y}\in\mathbb{N}^D} \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\, \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\, \mathbf{y}^i$$

$$= \mathbf{x}^i - \frac{1}{p_t(\mathbf{x})} \sum_{\mathbf{x}_0\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \sum_{\mathbf{y}\in\mathbb{N}^D} \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\, \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\, (\mathbf{x}^i-\mathbf{y}^i)$$

$$\overset{15}{=} \mathbf{x}^i - \frac{\mathrm{u}^i_t}{p_t(\mathbf{x})} \sum_{\mathbf{x}_0\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \sum_{\mathbf{y}\in\mathbb{N}^D} \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\, \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y}-e_i)$$

$$= \mathbf{x}^i - \mathrm{u}^i_t \frac{p_t(\mathbf{x}-e_i)}{p_t(\mathbf{x})} \frac{1}{p_t(\mathbf{x}-e_i)} \sum_{\mathbf{x}_0\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \sum_{\mathbf{y}\in\mathbb{N}^D} \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\, \Pi_{\mathrm{u}_t}(\mathbf{x}-e_i-\mathbf{y})$$

$$\overset{32}{=} \mathbf{x}^i - \mathrm{u}^i_t \frac{p_t(\mathbf{x}-e_i)}{p_t(\mathbf{x})} \sum_{\mathbf{y}\in\mathbb{N}^D} p_{\mathrm{B}_t|X_t=\mathbf{x}-e_i}(\mathbf{y}) = \mathbf{x}^i - \mathrm{u}^i_t \frac{p_t(\mathbf{x}-e_i)}{p_t(\mathbf{x})}.$$

□

The next result will help us show how to relate the `latent-up`, `latent-down`, and `data` denoisers.

Lemma B.4 (Relation between `data` and `latent` denoisers)**.** *For* $\mathbf{x}\in\mathbb{N}^D, t\in(0,\tau)$*, and* $i\in[D]$*,*

$$\mathrm{m}^i_{\texttt{data}}(\mathbf{x},t) = \mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t) + \frac{1-l^i_t}{l^i_t} \frac{p_t(\mathbf{x}+e_i)}{p_t(\mathbf{x})} \mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t) \tag{40}$$

$$= \mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t) + \frac{\mathrm{u}^i_t(1-l^i_t)}{l^i_t} \frac{\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t)}{\mathbf{x}^i+1-\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t)}. \tag{41}$$

Proof.

$$\mathrm{m}^i_{\texttt{data}}(\mathbf{x},t) = \mathbb{E}[\mathbf{x}^i_{\text{data}}|X_t=\mathbf{x}] = \sum_{\mathbf{x}_0\in\mathbb{N}^D} \mathbf{x}^i_0\, p_{\mathbf{x}_{\text{data}}|X_t=\mathbf{x}}(\mathbf{x}_0)$$

$$\overset{36}{=} \frac{1}{p_t(\mathbf{x})} \sum_{\mathbf{x}_0\in\mathbb{N}^D} \mathbf{x}^i_0\, (\mathcal{B}_{\mathbf{x}_0,l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x})\, p_{\text{data}}(\mathbf{x}_0)$$

$$= \frac{1}{p_t(\mathbf{x})} \sum_{\mathbf{x}_0\in\mathbb{N}^D} \sum_{\mathbf{n}\in\mathbb{N}^D} \mathbf{x}^i_0\, \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{n})\, \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{n})\, p_{\text{data}}(\mathbf{x}_0)$$

$$\overset{18}{=} \frac{1}{p_t(\mathbf{x})} \sum_{\mathbf{x}_0\in\mathbb{N}^D} \sum_{\mathbf{n}\in\mathbb{N}^D} \left(\frac{1-l^i_t}{l^i_t} (\mathbf{n}^i+1) \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{n}+e_i) + \mathbf{n}^i \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{n}) \right) \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{n})\, p_{\text{data}}(\mathbf{x}_0)$$

$$= \frac{1-l^i_t}{l^i_t} \frac{p_t(\mathbf{x}+e_i)}{p_t(\mathbf{x})} \sum_{\mathbf{n}\in\mathbb{N}^D} (\mathbf{n}^i+1) \frac{\sum_{\mathbf{x}_0\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{n}+e_i)}{p_t(\mathbf{x}+e_i)} \Pi_{\mathrm{u}_t}(\mathbf{x}+e_i-(\mathbf{n}+e_i))$$

$$+ \sum_{\mathbf{n}\in\mathbb{N}^D} \mathbf{n}^i \frac{\sum_{\mathbf{x}_0\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{n})}{p_t(\mathbf{x})} \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{n})$$

$$\overset{32}{=} \frac{1-l^i_t}{l^i_t} \frac{p_t(\mathbf{x}+e_i)}{p_t(\mathbf{x})} \sum_{\mathbf{n}\in\mathbb{N}^D} (\mathbf{n}^i+1)\, p_{\mathrm{B}_t|X_t=\mathbf{x}+e_i}(\mathbf{n}+e_i) + \sum_{\mathbf{n}\in\mathbb{N}^D} \mathbf{n}^i\, p_{\mathrm{B}_t|X_t=\mathbf{x}}(\mathbf{n})$$

$$\overset{\text{re-index}}{=} \frac{1-l^i_t}{l^i_t} \frac{p_t(\mathbf{x}+e_i)}{p_t(\mathbf{x})} \sum_{\mathbf{n}\in\mathbb{N}^D} \mathbf{n}^i\, p_{\mathrm{B}_t|X_t=\mathbf{x}+e_i}(\mathbf{n}) + \sum_{\mathbf{n}\in\mathbb{N}^D} \mathbf{n}^i\, p_{\mathrm{B}_t|X_t=\mathbf{x}}(\mathbf{n})$$

$$= \frac{1-l^i_t}{l^i_t} \frac{p_t(\mathbf{x}+e_i)}{p_t(\mathbf{x})} \mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t) + \mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t)$$

$$\overset{39}{=} \frac{\mathrm{u}^i_t(1-l^i_t)}{l^i_t} \frac{\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t)}{\mathbf{x}^i+1-\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t)} + \mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t).$$

□

We can now show that the `latent-up` and `latent-down` denoisers can be expressed in terms of each other, in a non-linear and non-local way. Consequently, the `data` denoiser can be expressed in terms of a single latent denoiser, in a non-linear and non-local way.

Corollary B.5 (Relations between `latent-up`, `latent-down`, and `data` denoisers)**.** *Fix* $\mathbf{x} \in \mathbb{N}^D, t \in (0, \tau)$ *and* $i \in [D]$*. Then,*

$$\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, t) = \frac{\mathrm{u}^i_t(1 - l^i_t)}{l^i_t} \frac{\mathbf{x}^i + 1 - \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x} + e_i, t)}{\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x} + e_i, t)}, \tag{42}$$

and

$$\mathrm{m}^i_{\texttt{data}}(\mathbf{x}, t) = \mathbf{x}^i - \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) + \frac{\mathrm{u}^i_t(1 - l^i_t)}{l^i_t} \frac{\mathbf{x}^i + 1 - \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x} + e_i, t)}{\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x} + e_i, t)}. \tag{43}$$

Proof. We have

$$\begin{aligned}
&\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) \overset{30}{=} \mathbf{x}^i - \mathrm{m}^i_{\texttt{latent}}(\mathbf{x}, t) \overset{29}{=} \mathbf{x}^i + \mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, t) - \mathrm{m}^i_{\texttt{data}}(\mathbf{x}, t) \\
&\overset{41}{=} \mathbf{x}^i + \mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, t) - \left(\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}, t) + \frac{\mathrm{u}^i_t(1 - l^i_t)}{l^i_t} \frac{\mathrm{m}^i_{\texttt{latent}}(\mathbf{x} + e_i, t)}{\mathbf{x}^i + 1 - \mathrm{m}^i_{\texttt{latent}}(\mathbf{x} + e_i, t)} \right) \\
&\overset{30}{=} \mathbf{x}^i + \mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, t) - \left(\mathbf{x}^i - \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) + \frac{\mathrm{u}^i_t(1 - l^i_t)}{l^i_t} \frac{\mathbf{x}^i + 1 - \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x} + e_i, t)}{\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x} + e_i, t)} \right),
\end{aligned}$$

so Equation (42) follows. For Equation (43) we use

$$\mathrm{m}_{\texttt{data}}(\mathbf{x}, t) = \mathbf{x} + \mathrm{m}_{\texttt{latent-down}}(\mathbf{x}, t) - \mathrm{m}_{\texttt{latent-up}}(\mathbf{x}, t),$$

together with Equation (42). □

We conclude by expressing the latent denoisers in terms of the marginal posteriors of the data given its noisy observations.

Lemma B.6. *Fix* $i \in [D]$ *and* $\mathbf{x} \in \mathbb{N}^D$*. Then,*

$$\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}, t) = \sum_{\mathbf{z}^i \in \mathbb{N}} p_{\mathbf{x}^i_{\text{data}} | X_t = \mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i \in \mathbb{N}} \mathbf{y}^i \frac{\mathcal{B}_{\mathbf{z}^i, l^i_t}(\mathbf{y}^i) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i - \mathbf{y}^i)}{(\mathcal{B}_{\mathbf{z}^i, l^i_t} * \Pi_{\mathrm{u}^i_t})(\mathbf{x}^i)}, \tag{44}$$

$$\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, t) = \sum_{\mathbf{z}^i \in \mathbb{N}} p_{\mathbf{x}^i_{\text{data}} | X_t = \mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i \in \mathbb{N}} \mathbf{y}^i \frac{\mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(\mathbf{y}^i) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i + \mathbf{y}^i - \mathbf{z}^i)}{\sum_{w \in \mathbb{N}} \mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(w) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i + w - \mathbf{z}^i)}, \tag{45}$$

$$\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) = \sum_{\mathbf{z}^i \in \mathbb{N}} p_{\mathbf{x}^i_{\text{data}} | X_t = \mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i \in \mathbb{N}} \mathbf{y}^i \frac{\mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(\mathbf{z}^i + \mathbf{y}^i - \mathbf{x}^i) \Pi_{\mathrm{u}^i_t}(\mathbf{y}^i)}{\sum_{w \in \mathbb{N}} \mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(\mathbf{z}^i + w - \mathbf{x}^i) \Pi_{\mathrm{u}^i_t}(w)}. \tag{46}$$

Proof. We first note that Equation (33), Equation (34), and Equation (35) show that $p_{\mathrm{B}_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y})$, $p_{\mathrm{L}_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y})$, and $p_{\mathrm{N}_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y})$ are product distributions with marginals

$$p_{\mathrm{B}^i_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}^i) = \frac{\mathcal{B}_{\mathbf{z}^i, l^i_t}(\mathbf{y}^i) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i - \mathbf{y}^i)}{(\mathcal{B}_{\mathbf{z}^i, l^i_t} * \Pi_{\mathrm{u}^i_t})(\mathbf{x}^i)}, \tag{47}$$

$$p_{\mathrm{L}^i_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}^i) = \frac{\mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(\mathbf{y}^i) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i + \mathbf{y}^i - \mathbf{z}^i)}{\sum_{w \in \mathbb{N}} \mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(w) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i + w - \mathbf{z}^i)}, \tag{48}$$

$$p_{\mathrm{N}^i_t | X_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}^i) = \frac{\mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(\mathbf{z}^i + \mathbf{y}^i - \mathbf{x}^i) \Pi_{\mathrm{u}^i_t}(\mathbf{y}^i)}{\sum_{w \in \mathbb{N}} \mathcal{B}_{\mathbf{z}^i, \mathrm{d}^i_t}(\mathbf{z}^i + w - \mathbf{x}^i) \Pi_{\mathrm{u}^i_t}(w)}. \tag{49}$$

By marginalization it follows that

$$
\begin{aligned}
p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i) &= \sum_{\mathbf{z}^{-i}\in\mathbb{N}^{D-1}} p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i,\mathbf{x}_{\text{data}}^{-i}=\mathbf{z}^{-i}}(\mathbf{y}^i)\, p_{\mathbf{x}_{\text{data}}^{-i}|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{z}^{-i}) \\
&= \sum_{\mathbf{z}^{-i}\in\mathbb{N}^{D-1}} p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i)\, p_{\mathbf{x}_{\text{data}}^{-i}|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{z}^{-i}) \\
&= p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i) \underbrace{\sum_{\mathbf{z}^{-i}\in\mathbb{N}^{D-1}} p_{\mathbf{x}_{\text{data}}^{-i}|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{z}^{-i})}_{=1} = p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i),
\end{aligned}
$$

and similarly,

$$
\begin{aligned}
p_{\mathrm{L}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i) &= p_{\mathrm{L}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i), \\
p_{\mathrm{N}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i) &= p_{\mathrm{N}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i).
\end{aligned}
$$

Combining with Equation (47), Equation (48), and Equation (49) we conclude

$$
p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i) = \frac{\mathcal{B}_{\mathbf{z}^i,l_t^i}(\mathbf{y}^i)\Pi_{\mathrm{u}_t^i}(\mathbf{x}^i-\mathbf{y}^i)}{(\mathcal{B}_{\mathbf{z}^i,l_t^i} * \Pi_{\mathrm{u}_t^i})(\mathbf{x}^i)}, \tag{50}
$$

$$
p_{\mathrm{L}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i) = \frac{\mathcal{B}_{\mathbf{z}^i,\mathrm{d}_t^i}(\mathbf{y}^i)\Pi_{\mathrm{u}_t^i}(\mathbf{x}^i+\mathbf{y}^i-\mathbf{z}^i)}{\sum_{w\in\mathbb{N}}\mathcal{B}_{\mathbf{z}^i,\mathrm{d}_t^i}(w)\Pi_{\mathrm{u}_t^i}(\mathbf{x}^i+w-\mathbf{z}^i)}, \tag{51}
$$

$$
p_{\mathrm{N}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i) = \frac{\mathcal{B}_{\mathbf{z}^i,\mathrm{d}_t^i}(\mathbf{z}^i+\mathbf{y}^i-\mathbf{x}^i)\Pi_{\mathrm{u}_t^i}(\mathbf{y}^i)}{\sum_{w\in\mathbb{N}}\mathcal{B}_{\mathbf{z}^i,\mathrm{d}_t^i}(\mathbf{z}^i+w-\mathbf{x}^i)\Pi_{\mathrm{u}_t^i}(w)}. \tag{52}
$$

Hence, by Bayes' rule,

$$
\begin{aligned}
\mathrm{m}_{\texttt{latent}}^i(\mathbf{x},t) &= \mathbb{E}[\mathrm{B}_t^i|X_t=\mathbf{x}] = \sum_{\mathbf{y}^i\in\mathbb{N}} \mathbf{y}^i\, p_{\mathrm{B}_t^i|X_t=\mathbf{x}}(\mathbf{y}^i) \\
&= \sum_{\mathbf{y}^i\in\mathbb{N}} \mathbf{y}^i\, \frac{p_{\mathrm{B}_t^i,X_t}(\mathbf{y}^i,\mathbf{x})}{p_{X_t}(\mathbf{x})} = \sum_{\mathbf{y}^i\in\mathbb{N}} \mathbf{y}^i\, \frac{1}{p_{X_t}(\mathbf{x})} \sum_{\mathbf{z}^i\in\mathbb{N}} p_{\mathrm{B}_t^i,X_t,\mathbf{x}_{\text{data}}^i}(\mathbf{y}^i,\mathbf{x},\mathbf{z}^i) \\
&= \sum_{\mathbf{y}^i\in\mathbb{N}} \mathbf{y}^i \sum_{\mathbf{z}^i\in\mathbb{N}} p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i)\frac{p_{X_t,\mathbf{x}_{\text{data}}^i}(\mathbf{x},\mathbf{z}^i)}{p_{X_t}(\mathbf{x})} \\
&= \sum_{\mathbf{y}^i\in\mathbb{N}} \mathbf{y}^i \sum_{\mathbf{z}^i\in\mathbb{N}} p_{\mathrm{B}_t^i|X_t=\mathbf{x},\mathbf{x}_{\text{data}}^i=\mathbf{z}^i}(\mathbf{y}^i) p_{\mathbf{x}_{\text{data}}^i|X_t=\mathbf{x}}(\mathbf{z}^i) \\
&\overset{50}{=} \sum_{\mathbf{z}^i\in\mathbb{N}} p_{\mathbf{x}_{\text{data}}^i|X_t=\mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i\in\mathbb{N}} \mathbf{y}^i\, \frac{\mathcal{B}_{\mathbf{z}^i,l_t^i}(\mathbf{y}^i)\Pi_{\mathrm{u}_t^i}(\mathbf{x}^i-\mathbf{y}^i)}{(\mathcal{B}_{\mathbf{z}^i,l_t^i} * \Pi_{\mathrm{u}_t^i})(\mathbf{x}^i)},
\end{aligned}
$$

which establishes Equation (44). Similarly, using Equation (51) and Equation (52), we get Equation (45) and Equation (46). □

B.2 Forward and backward processes

In this section we will show how to use the latent denoisers in order to define forward and backward processes with the marginals of the Binomial-Poisson perturbation (27). Consequently, we prove Theorem B.1.

For $i \in [D]$ and $t \in [0,\tau]$ set

$$
\mathrm{u}_0^i = 0, \quad l_0^i = 1, \quad l_\tau^i = 0, \qquad \dot{\mathrm{u}}_t^i := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{u}_t^i, \quad \dot{l}_t^i := \frac{\mathrm{d}}{\mathrm{d}t} l_t^i. \tag{53}
$$

We begin by deriving the evolution equation of the marginals of the Binomial-Poisson perturbation (27).

Lemma B.7. *Let $X_t \sim p_t$ where X_t is as in Equation (27). Then, for $\mathbf{x} \in \mathbb{N}^D$ and $t \in (0, \tau)$,*

$$\partial_t p_t(\mathbf{x}) = \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{1-l_t^i} \mathrm{m}_{\texttt{data}}^i(\mathbf{x}, t) \right) p_t(\mathbf{x}) - \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i \mathrm{u}_t^i}{l_t^i(1-l_t^i)} - \dot{\mathrm{u}}_t^i \right) p_t(\mathbf{x} - e_i) \tag{54}$$

$$= \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{1-l_t^i} \mathrm{m}_{\texttt{latent}}^i(\mathbf{x}, t) \right) p_t(\mathbf{x}) - \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i \mathrm{u}_t^i}{l_t^i(1-l_t^i)} - \dot{\mathrm{u}}_t^i \right) p_t(\mathbf{x} - e_i) \tag{55}$$

$$- \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i} \mathrm{m}_{\texttt{latent}}^i(\mathbf{x} + e_i, t) \right) p_t(\mathbf{x} + e_i).$$

Proof.

$$\partial_t p_t(\mathbf{x}) = \partial_t \sum_{\mathbf{x}_0 \in \mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)(\mathcal{B}_{\mathbf{x}_0, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x})$$

$$\overset{20}{=} \sum_{\mathbf{x}_0 \in \mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \left\{ (\mathcal{B}_{\mathbf{x}_0, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x}) \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{1-l_t^i} \mathbf{x}_0^i \right) \right\}$$

$$- \sum_{\mathbf{x}_0 \in \mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0) \left\{ \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i \mathrm{u}_t^i}{l_t^i(1-l_t^i)} - \dot{\mathrm{u}}_t^i \right) (\mathcal{B}_{\mathbf{x}_0, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x} - e_i) \right\}$$

$$= p_t(\mathbf{x}) \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i \right) - \sum_{i=1}^{D} \frac{\dot{l}_t^i}{1-l_t^i} \sum_{\mathbf{x}_0 \in \mathbb{N}^D} \mathbf{x}_0^i \underbrace{p_{\text{data}}(\mathbf{x}_0)(\mathcal{B}_{\mathbf{x}_0, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x})}_{= p_{0,t}(\mathbf{x}_0, \mathbf{x}) = p_{0|t}(\mathbf{x}_0|\mathbf{x}) p_t(\mathbf{x})}$$

$$- \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i \mathrm{u}_t^i}{l_t^i(1-l_t^i)} - \dot{\mathrm{u}}_t^i \right) \underbrace{\sum_{\mathbf{x}_0 \in \mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)(\mathcal{B}_{\mathbf{x}_0, l_t} * \Pi_{\mathrm{u}_t})(\mathbf{x} - e_i)}_{p_t(\mathbf{x} - e_i)}$$

$$= p_t(\mathbf{x}) \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{1-l_t^i} \sum_{\mathbf{x}_0 \in \mathbb{N}^D} \mathbf{x}_0^i \, p_{0|t}(\mathbf{x}_0|\mathbf{x}) \right) - \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i \mathrm{u}_t^i}{l_t^i(1-l_t^i)} - \dot{\mathrm{u}}_t^i \right) p_t(\mathbf{x} - e_i)$$

$$= p_t(\mathbf{x}) \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{1-l_t^i} \mathrm{m}_{\texttt{data}}^i(\mathbf{x}, t) \right) - \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i \mathrm{u}_t^i}{l_t^i(1-l_t^i)} - \dot{\mathrm{u}}_t^i \right) p_t(\mathbf{x} - e_i),$$

and we conclude as

$$p_t(\mathbf{x}) \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{1-l_t^i} \mathrm{m}_{\texttt{data}}^i(\mathbf{x}, t) \right)$$

$$= p_t(\mathbf{x}) \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i(1-l_t^i)} \mathbf{x}^i - \dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{1-l_t^i} \mathrm{m}_{\texttt{latent}}^i(\mathbf{x}, t) \right) - \sum_{i=1}^{D} \left(\frac{\dot{l}_t^i}{l_t^i} \mathrm{m}_{\texttt{latent}}^i(\mathbf{x} + e_i, t) \right) p_t(\mathbf{x} + e_i).$$

□

Next we will show how to use the denoiser (38) to define a CTMC whose marginals are the same as the Binomial-Poisson perturbation (27).

Proposition B.8. *Let $(\overrightarrow{\mathbf{X}_t})_{t \in [0,\tau]}$ be the CTMC with, for $0 < t < \tau$, rate matrices*

$$\overrightarrow{Q}_{\mathbf{xy}}(t) = \begin{cases} \dot{\mathrm{u}}_t^i & \text{if} \quad \mathbf{y} = \mathbf{x} + e_i, \qquad i \in [D] \\ -\frac{\dot{l}_t^i}{l_t^i} \mathrm{m}_{\texttt{latent}}^i(\mathbf{x}, t) & \text{if} \quad \mathbf{y} = \mathbf{x} - e_i, \qquad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\} \\ -\sum_{\mathbf{y} \in \{\mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\}} \overrightarrow{Q}_{\mathbf{xy}}(t) & \text{if} \quad \mathbf{y} = \mathbf{x} \end{cases}, \tag{56}$$

with $\overrightarrow{\mathbf{X}_0} \sim p_{\text{data}}$. *Then,*

$$\overrightarrow{\mathbf{X}_t} \sim p_t \qquad \text{for all} \qquad t \in [0, \tau].$$

Proof. It suffices to show that $(p_t)_{t\in[0,\tau]}$ satisfies the Kolmogorov equation

$$\begin{aligned}
\partial_t p_t(\mathbf{x}) &= \sum_{\mathbf{y}\in\mathbb{N}^D} p_t(\mathbf{y})\overrightarrow{Q}_{\mathbf{yx}}(t) \\
&= \sum_{i=1}^{D} \left\{ \dot{\mathrm{u}}_t^i p_t(\mathbf{x}-e_i) - \frac{\dot{l}_t^i}{l_t^i}\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t)p_t(\mathbf{x}+e_i) - \left(\dot{\mathrm{u}}_t^i - \frac{\dot{l}_t^i}{l_t^i}\mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t)\right) p_t(\mathbf{x}) \right\}.
\end{aligned} \tag{57}$$

To establish (57) write

$$\begin{aligned}
\partial_t p_t(\mathbf{x}) &= \partial_t \sum_{\mathbf{x}_0\in\mathbb{N}^D}\sum_{\mathbf{y}\in\mathbb{N}^D} \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\, p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y}) \\
&= \sum_{\mathbf{x}_0\in\mathbb{N}^D}\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)\left\{\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\partial_t\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y}) + \mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\partial_t\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\right\} \\
&\overset{19}{=} \sum_{\mathbf{x}_0\in\mathbb{N}^D}\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\sum_{i=1}^{D}\frac{\dot{l}_t^i}{l_t^i}\left(\mathbf{y}^i\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y}) - (\mathbf{y}^i+1)\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y}+e_i)\right) \\
&\overset{16}{+} \sum_{\mathbf{x}_0\in\mathbb{N}^D}\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\sum_{i=1}^{D}\dot{\mathrm{u}}_t^i\left(\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y}-e_i) - \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\right) \\
&=: B + N.
\end{aligned}$$

We now analyze the two terms B and N. We have

$$\begin{aligned}
B &= \sum_{i=1}^{D}\frac{\dot{l}_t^i}{l_t^i}\sum_{\mathbf{x}_0\in\mathbb{N}^D}\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\mathbf{y}^i\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y}) \\
&\quad - \sum_{i=1}^{D}\frac{\dot{l}_t^i}{l_t^i}\sum_{\mathbf{x}_0\in\mathbb{N}^D}\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})(\mathbf{y}^i+1)\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y}+e_i) \\
&= p_t(\mathbf{x})\sum_{i=1}^{D}\frac{\dot{l}_t^i}{l_t^i}\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\mathrm{B}_t|X_t=\mathbf{x}}(\mathbf{y})\mathbf{y}^i - \sum_{i=1}^{D}\frac{\dot{l}_t^i}{l_t^i}p_t(\mathbf{x}+e_i)\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\mathrm{B}_t|X_t=\mathbf{x}+e_i}(\mathbf{y})\mathbf{y}^i \\
&= \sum_{i=1}^{D}\frac{\dot{l}_t^i}{l_t^i}\left[p_t(\mathbf{x})\mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t) - p_t(\mathbf{x}+e_i)\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t)\right],
\end{aligned}$$

while

$$N = \sum_{\mathbf{x}_0\in\mathbb{N}^D}\sum_{\mathbf{y}\in\mathbb{N}^D} p_{\text{data}}(\mathbf{x}_0)\mathcal{B}_{\mathbf{x}_0,l_t}(\mathbf{y})\sum_{i=1}^{D}\dot{\mathrm{u}}_t^i\left(\Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y}-e_i) - \Pi_{\mathrm{u}_t}(\mathbf{x}-\mathbf{y})\right) = \sum_{i=1}^{D}\dot{\mathrm{u}}_t^i\left(p_t(\mathbf{x}-e_i) - p_t(\mathbf{x})\right).$$

It follows that

$$\partial_t p_t(\mathbf{x}) = \sum_{i=1}^{D}\left\{\frac{\dot{l}_t^i}{l_t^i}[p_t(\mathbf{x})\mathrm{m}^i_{\texttt{latent}}(\mathbf{x},t) - p_t(\mathbf{x}+e_i)\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,t)] + \dot{\mathrm{u}}_t^i\left[p_t(\mathbf{x}-e_i) - p_t(\mathbf{x})\right]\right\},$$

which is exactly Equation (57). □

We can now derive the rates of the time-reversal of $(\overrightarrow{\mathbf{X}_t})_{t\in[0,\tau]}$.

Corollary B.9. *With $s = \tau - t$, the time reversal $(\overleftarrow{\mathbf{X}_s})_{s\in[0,\tau]}$ of $(\overrightarrow{\mathbf{X}_t})_{t\in[0,\tau]}$ has, for $0 < s < \tau$, rate matrices*

$$\overleftarrow{Q}_{\mathbf{xy}}(s) = \begin{cases} \frac{\dot{\mathrm{u}}^i_{\tau-s}}{\mathrm{u}^i_{\tau-s}}(\mathbf{x}^i - \mathrm{m}^i_{\texttt{latent}}(\mathbf{x}, \tau-s)) & \text{if} \quad \mathbf{y} = \mathbf{x} - e_i, \qquad i \in [D] \\ -\frac{\dot{l}^i_{\tau-s}\mathrm{u}^i_{\tau-s}}{l^i_{\tau-s}} \frac{\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,\tau-s)}{\mathbf{x}^i+1-\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,\tau-s)} & \text{if} \quad \mathbf{y} = \mathbf{x} + e_i, \qquad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\} \\ -\sum_{\mathbf{y}\in\{\mathbf{x}\pm e_1,\ldots,\mathbf{x}\pm e_D\}} \overleftarrow{Q}_{\mathbf{xy}}(s) & \text{if} \quad \mathbf{y} = \mathbf{x} \end{cases}. \tag{58}$$

Proof. By the time-reversal formula

$$\overleftarrow{Q}_{\mathbf{xy}}(s) = \frac{p_{\tau-s}(\mathbf{y})}{p_{\tau-s}(\mathbf{x})} \overrightarrow{Q}_{\mathbf{yx}}(\tau - s),$$

and Equation (39), we have

$$\overleftarrow{Q}_{\mathbf{x},\mathbf{x}-e_i}(s) = \frac{p_{\tau-s}(\mathbf{x}-e_i)}{p_{\tau-s}(\mathbf{x})}\overrightarrow{Q}_{\mathbf{x}-e_i,\mathbf{x}}(\tau-s) = \frac{\mathbf{x}^i - \mathrm{m}^i_{\texttt{latent}}(\mathbf{x},\tau-s)}{\mathrm{u}^i_{\tau-s}}\dot{\mathrm{u}}^i_{\tau-s},$$

while

$$\begin{aligned} \overleftarrow{Q}_{\mathbf{x},\mathbf{x}+e_i}(s) &= \frac{p_{\tau-s}(\mathbf{x}+e_i)}{p_{\tau-s}(\mathbf{x})}\overrightarrow{Q}_{\mathbf{x}+e_i,\mathbf{x}}(\tau-s) \\ &= -\frac{\mathrm{u}^i_{\tau-s}}{\mathbf{x}^i + 1 - \mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,\tau-s)}\frac{\dot{l}^i_{\tau-s}}{l^i_{\tau-s}}\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,\tau-s) \\ &= -\dot{l}^i_{\tau-s}\frac{\mathrm{u}^i_{\tau-s}}{l^i_{\tau-s}}\frac{\mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,\tau-s)}{\mathbf{x}^i + 1 - \mathrm{m}^i_{\texttt{latent}}(\mathbf{x}+e_i,\tau-s)}. \end{aligned}$$

□

B.3 Proof of Theorem B.1

The time-reversed chain in Corollary B.9, indexed by sampling time s, starts from $p_\tau = \mathrm{Poisson}(\mathrm{u}_\tau)$ and has marginals $p_{\tau-s}$. We use Equation (29)-Equation (30), together with Equation (41) and $\mathrm{d}_t = 1 - l_t$, to express its rates in terms of the `latent-up` and `latent-down` denoisers, yielding Equation (24).

C The Poisson-Poisson process

The main result of this section is the Poisson-Poisson part of Theorem 2.1:

Theorem C.1 (The Poisson-Poisson process)**.** *Under the schedule assumptions of Theorem 2.1, set the noise schedules $\mathrm{d}_t, \mathrm{u}_t \in \mathbb{R}^D_{\geq 0}$ to satisfy*

$$\mathrm{d}^i_0 = 0, \quad \mathrm{u}^i_0 = 0, \qquad \dot{\mathrm{u}}^i_t := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{u}^i_t, \quad \dot{\mathrm{d}}^i_t := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{d}^i_t, \qquad i \in [D]. \tag{59}$$

Let $(q_t)_{t\in[0,\tau]}$ be the marginals of $(Y_t)_{t\in[0,\tau]}$,

$$Y_t := \mathbf{x}_{\text{data}} - \mathrm{L}_t + \mathrm{N}_t, \qquad \text{with} \qquad \mathrm{L}_t \sim \mathrm{Poisson}(\mathrm{d}_t), \quad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{60}$$

where $(\mathbf{x}_{\text{data}}, \mathrm{L}_t, \mathrm{N}_t)$ are independent, and define the latent denoisers

$$\mathrm{m}_{\texttt{latent-down}}(\mathbf{x}, t) := \mathbb{E}[\mathrm{L}_t | Y_t = \mathbf{x}], \qquad \mathrm{m}_{\texttt{latent-up}}(\mathbf{x}, t) := \mathbb{E}[\mathrm{N}_t | Y_t = \mathbf{x}]. \tag{61}$$

Write $s = \tau - t$ for reverse time. The time reversal $(\mathbf{Y}_s)_{s\in[0,\tau]}$ of the forward CTMC in Proposition C.7 has initial distribution $\mathbf{Y}_0 \sim q_\tau$ and, for $0 < s < \tau$, rate matrices

$$Q_{\mathbf{xy}}(s) = \begin{cases} \frac{\dot{\mathrm{u}}^i_{\tau-s}}{\mathrm{u}^i_{\tau-s}}\,\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, \tau-s) & \text{if} \quad \mathbf{y} = \mathbf{x} - e_i, \qquad i \in [D] \\ \frac{\dot{\mathrm{d}}^i_{\tau-s}}{\mathrm{d}^i_{\tau-s}}\,\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, \tau-s) & \text{if} \quad \mathbf{y} = \mathbf{x} + e_i, \qquad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\} \\ -\sum_{\mathbf{y}\in\{\mathbf{x}\pm e_1,\ldots,\mathbf{x}\pm e_D\}} Q_{\mathbf{xy}}(s) & \text{if} \quad \mathbf{y} = \mathbf{x} \end{cases}. \tag{62}$$

In particular,

$$\mathbf{Y}_s \sim q_{\tau-s} \qquad \text{for all} \qquad s \in [0, \tau]. \tag{63}$$

To avoid confusion with the Binomial-Poisson process, in this section we change the notation according to

$$\mathrm{L}_t \mapsto \mathrm{D}_t, \qquad \mathrm{N}_t \mapsto \mathrm{U}_t, \qquad \mathrm{d}_t \mapsto l_t, \qquad \dot{\mathrm{d}}_t \mapsto \dot{l}_t, \tag{64}$$

so we focus on the perturbations of $\mathbf{x}_{\text{data}} \sim p_{\text{data}}$, for a given $t \in [0, \tau]$, by

$$Y_t := \mathbf{x}_{\text{data}} - \mathrm{D}_t + \mathrm{U}_t, \qquad \text{with} \qquad \mathrm{D}_t \sim \mathrm{Poisson}(l_t), \quad \mathrm{U}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{65}$$

where $(\mathbf{x}_{\text{data}}, \mathrm{D}_t, \mathrm{U}_t)$ are independent. We begin by recording some of the conditional distributions of $(\mathbf{x}_{\text{data}}, \mathrm{D}_t, \mathrm{U}_t)$.

Lemma C.2. *Let q_t be the distribution of Y_t given in* (65)*. For $\mathbf{x} \in \mathbb{Z}^D$,*

$$\begin{aligned} q_t(\mathbf{x}) &= \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{y})\, \Pi_{\mathrm{u}_t}(\mathbf{x} + \mathbf{y} - \mathbf{x}_0) \\ &= \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{y})\, \Pi_{l_t}(\mathbf{y} + \mathbf{x}_0 - \mathbf{x}). \end{aligned} \tag{66}$$

Fix $\mathbf{x} \in \mathbb{Z}^D$, then, for $\mathbf{y} \in \mathbb{Z}^D$,

$$q_{\mathrm{D}_t | Y_t = \mathbf{x}}(\mathbf{y}) = \frac{\Pi_{l_t}(\mathbf{y})}{q_t(\mathbf{x})} \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{x} + \mathbf{y} - \mathbf{x}_0), \tag{67}$$

$$q_{\mathrm{U}_t | Y_t = \mathbf{x}}(\mathbf{y}) = \frac{\Pi_{\mathrm{u}_t}(\mathbf{y})}{q_t(\mathbf{x})} \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{x}_0 - \mathbf{x} + \mathbf{y}). \tag{68}$$

Fix $\mathbf{x}, \mathbf{z} \in \mathbb{Z}^D$, then, for $\mathbf{y} \in \mathbb{Z}^D$,

$$q_{\mathrm{D}_t | Y_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}) = \frac{\Pi_{l_t}(\mathbf{y}) \Pi_{\mathrm{u}_t}(\mathbf{x} + \mathbf{y} - \mathbf{z})}{\sum_{\tilde{\mathbf{y}} \in \mathbb{Z}^D} \Pi_{l_t}(\tilde{\mathbf{y}}) \Pi_{\mathrm{u}_t}(\mathbf{x} + \tilde{\mathbf{y}} - \mathbf{z})}, \tag{69}$$

$$q_{\mathrm{U}_t | Y_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}) = \frac{\Pi_{\mathrm{u}_t}(\mathbf{y}) \Pi_{l_t}(\mathbf{y} + \mathbf{z} - \mathbf{x})}{\sum_{\tilde{\mathbf{y}} \in \mathbb{Z}^D} \Pi_{\mathrm{u}_t}(\tilde{\mathbf{y}}) \Pi_{l_t}(\tilde{\mathbf{y}} + \mathbf{z} - \mathbf{x})}. \tag{70}$$

Fix $\mathbf{y} \in \mathbb{Z}^D$, then, for $\mathbf{x} \in \mathbb{Z}^D$,

$$q_{\mathbf{x}_{\text{data}} | Y_t = \mathbf{y}}(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{q_t(\mathbf{y})} (\Pi_{l_t}(\mathbf{x} - \cdot) * \Pi_{\mathrm{u}_t})(\mathbf{y}). \tag{71}$$

Proof. Equation (66) holds by the definition (65). For Equation (67) and Equation (68) we use Bayes' rule to get

$$\begin{aligned} q_{\mathrm{D}_t | Y_t = \mathbf{x}}(\mathbf{y}) &= \frac{q_{\mathrm{D}_t, Y_t}(\mathbf{y}, \mathbf{x})}{q_{Y_t}(\mathbf{x})} = \frac{q_{\mathrm{D}_t, \mathbf{x}_{\text{data}} + \mathrm{U}_t}(\mathbf{y}, \mathbf{x} + \mathbf{y})}{q_t(\mathbf{x})} = \frac{q_{\mathrm{D}_t}(\mathbf{y}) q_{\mathbf{x}_{\text{data}} + \mathrm{U}_t}(\mathbf{x} + \mathbf{y})}{q_t(\mathbf{x})} \\ &= \frac{\Pi_{l_t}(\mathbf{y})}{q_t(\mathbf{x})} \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{x} + \mathbf{y} - \mathbf{x}_0), \end{aligned}$$

and

$$\begin{aligned} q_{\mathrm{U}_t | Y_t = \mathbf{x}}(\mathbf{y}) &= \frac{q_{\mathrm{U}_t, Y_t}(\mathbf{y}, \mathbf{x})}{q_{Y_t}(\mathbf{x})} = \frac{q_{\mathrm{U}_t, \mathbf{x}_{\text{data}} - \mathrm{D}_t}(\mathbf{y}, \mathbf{x} - \mathbf{y})}{q_t(\mathbf{x})} = \frac{q_{\mathrm{U}_t}(\mathbf{y}) q_{\mathbf{x}_{\text{data}} - \mathrm{D}_t}(\mathbf{x} - \mathbf{y})}{q_t(\mathbf{x})} \\ &= \frac{\Pi_{\mathrm{u}_t}(\mathbf{y})}{q_t(\mathbf{x})} \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{x}_0 - \mathbf{x} + \mathbf{y}). \end{aligned}$$

For Equation (69) and Equation (70) we use the independence of $(\mathbf{x}_{\text{data}}, \mathrm{D}_t, \mathrm{U}_t)$ and Bayes' rule to get

$$\begin{aligned} q_{\mathrm{D}_t | Y_t = \mathbf{x}, \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{y}) &= \frac{q_{\mathrm{D}_t, Y_t, \mathbf{x}_{\text{data}}}(\mathbf{y}, \mathbf{x}, \mathbf{z})}{q_{Y_t, \mathbf{x}_{\text{data}}}(\mathbf{x}, \mathbf{z})} = \frac{q_{\mathrm{D}_t, \mathrm{U}_t, \mathbf{x}_{\text{data}}}(\mathbf{y}, \mathbf{x} + \mathbf{y} - \mathbf{z}, \mathbf{z})}{q_{Y_t, \mathbf{x}_{\text{data}}}(\mathbf{x}, \mathbf{z})} \\ = \frac{q_{\mathrm{D}_t}(\mathbf{y}) q_{\mathrm{U}_t}(\mathbf{x} + \mathbf{y} - \mathbf{z}) p_{\text{data}}(\mathbf{z})}{q_{Y_t | \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{x}) p_{\text{data}}(\mathbf{z})} &= \frac{\Pi_{l_t}(\mathbf{y}) \Pi_{\mathrm{u}_t}(\mathbf{x} + \mathbf{y} - \mathbf{z})}{q_{Y_t | \mathbf{x}_{\text{data}} = \mathbf{z}}(\mathbf{x})} = \frac{\Pi_{l_t}(\mathbf{y}) \Pi_{\mathrm{u}_t}(\mathbf{x} + \mathbf{y} - \mathbf{z})}{\sum_{\tilde{\mathbf{y}} \in \mathbb{Z}^D} \Pi_{l_t}(\tilde{\mathbf{y}}) \Pi_{\mathrm{u}_t}(\mathbf{x} + \tilde{\mathbf{y}} - \mathbf{z})}, \end{aligned}$$

and similarly,

$$q_{\mathrm{U}_t|Y_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}) = \frac{q_{\mathrm{U}_t,Y_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{x},\mathbf{z})}{q_{Y_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})} = \frac{q_{\mathrm{U}_t,\mathrm{D}_t,\mathbf{x}_{\text{data}}}(\mathbf{y},\mathbf{y}+\mathbf{z}-\mathbf{x},\mathbf{z})}{q_{Y_t,\mathbf{x}_{\text{data}}}(\mathbf{x},\mathbf{z})}$$
$$= \frac{q_{\mathrm{U}_t}(\mathbf{y})q_{\mathrm{D}_t}(\mathbf{y}+\mathbf{z}-\mathbf{x})p_{\text{data}}(\mathbf{z})}{q_{Y_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{x})p_{\text{data}}(\mathbf{z})} = \frac{\Pi_{\mathrm{u}_t}(\mathbf{y})\Pi_{l_t}(\mathbf{y}+\mathbf{z}-\mathbf{x})}{q_{Y_t|\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{x})} = \frac{\Pi_{\mathrm{u}_t}(\mathbf{y})\Pi_{l_t}(\mathbf{y}+\mathbf{z}-\mathbf{x})}{\sum_{\tilde{\mathbf{y}}\in\mathbb{Z}^D}\Pi_{\mathrm{u}_t}(\tilde{\mathbf{y}})\Pi_{l_t}(\tilde{\mathbf{y}}+\mathbf{z}-\mathbf{x})}.$$

Equation (71) also follows from Bayes' rule. □

C.1 Discrete Tweedie's formula

In this section we derive a discrete Tweedie's formula for the Poisson-Poisson perturbation (65) . This formula is crucial in proving Theorem C.1.

Lemma C.3 (Discrete Tweedie's formula)**.** *Fix $t \in (0,\tau)$, and let*

$$Y_t := \mathbf{x}_{\text{data}} - \mathrm{D}_t + \mathrm{U}_t, \qquad \text{with} \qquad \mathrm{D}_t \sim \mathrm{Poisson}(l_t), \quad \mathrm{U}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \tag{72}$$

where $(\mathbf{x}_{\text{data}}, \mathrm{D}_t, \mathrm{U}_t)$ are independent. Denote by q_t the distribution of Y_t, i.e., $Y_t \sim q_t$, and define

$$\mathrm{m}_{\texttt{latent-down}}(\mathbf{x},t) := \mathbb{E}[\mathrm{D}_t|Y_t=\mathbf{x}], \qquad \mathrm{m}_{\texttt{latent-up}}(\mathbf{x},t) := \mathbb{E}[\mathrm{U}_t|Y_t=\mathbf{x}]. \tag{73}$$

Then,

$$\frac{q_t(\mathbf{x}+e_i)}{q_t(\mathbf{x})} = \frac{1}{l_t^i}\,\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x},t), \qquad \mathbf{x}\in\mathbb{Z}^D, \quad t\in(0,\tau), \quad i\in[D], \tag{74}$$

and

$$\frac{q_t(\mathbf{x}-e_i)}{q_t(\mathbf{x})} = \frac{\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x},t)}{\mathrm{u}_t^i}, \qquad \mathbf{x}\in\mathbb{Z}^D, \quad t\in(0,\tau), \quad i\in[D]. \tag{75}$$

Proof.

$$\begin{aligned}
&\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x},t) = \mathbb{E}[\mathrm{D}_t^i|Y_t=\mathbf{x}] = \sum_{\mathbf{y}\in\mathbb{Z}^D}\mathbf{y}^i q_{\mathrm{D}_t|Y_t=\mathbf{x}}(\mathbf{y})\\
&\overset{67}{=} \frac{1}{q_t(\mathbf{x})}\sum_{\mathbf{y}\in\mathbb{Z}^D}\mathbf{y}^i\Pi_{l_t}(\mathbf{y})\sum_{\mathbf{x}_0\in\mathbb{Z}^D}p_{\text{data}}(\mathbf{x}_0)\,\Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0)\\
&\overset{15}{=} \frac{1}{q_t(\mathbf{x})}\sum_{\mathbf{y}\in\mathbb{Z}^D}l_t^i\,\Pi_{l_t}(\mathbf{y}-e_i)\sum_{\mathbf{x}_0\in\mathbb{Z}^D}p_{\text{data}}(\mathbf{x}_0)\,\Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0)\\
&= \frac{l_t^i}{q_t(\mathbf{x})}\sum_{\mathbf{y}\in\mathbb{Z}^D}\sum_{\mathbf{x}_0\in\mathbb{Z}^D}p_{\text{data}}(\mathbf{x}_0)\,\Pi_{l_t}(\mathbf{y}-e_i)\,\Pi_{\mathrm{u}_t}(\mathbf{x}+(\mathbf{y}-e_i)-(\mathbf{x}_0-e_i))\\
&\overset{\text{re-index}}{=} \frac{l_t^i}{q_t(\mathbf{x})}\sum_{\mathbf{y}\in\mathbb{Z}^D}\sum_{\mathbf{x}_0\in\mathbb{Z}^D}p_{\text{data}}(\mathbf{x}_0)\,\Pi_{l_t}(\mathbf{y})\,\Pi_{\mathrm{u}_t}((\mathbf{x}+e_i)+\mathbf{y}-\mathbf{x}_0)\\
&\overset{66}{=} l_t^i\,\frac{q_t(\mathbf{x}+e_i)}{q_t(\mathbf{x})},
\end{aligned}$$

and similarly,

$$\begin{aligned}
\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) &= \mathbb{E}[\mathrm{U}^i_t | Y_t = \mathbf{x}] = \sum_{\mathbf{y} \in \mathbb{Z}^D} \mathbf{y}^i q_{\mathrm{U}_t | Y_t = \mathbf{x}}(\mathbf{y}) \\
&\overset{68}{=} \frac{1}{q_t(\mathbf{x})} \sum_{\mathbf{y} \in \mathbb{Z}^D} \mathbf{y}^i \Pi_{\mathrm{u}_t}(\mathbf{y}) \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{x}_0 - \mathbf{x} + \mathbf{y}) \\
&\overset{15}{=} \frac{1}{q_t(\mathbf{x})} \sum_{\mathbf{y} \in \mathbb{Z}^D} \mathrm{u}^i_t\, \Pi_{\mathrm{u}_t}(\mathbf{y} - e_i) \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{x}_0 - \mathbf{x} + \mathbf{y}) \\
&= \frac{\mathrm{u}^i_t}{q_t(\mathbf{x})} \sum_{\mathbf{y} \in \mathbb{Z}^D} \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{y} - e_i)\, \Pi_{l_t}(\mathbf{x}_0 - (\mathbf{x} - e_i) + (\mathbf{y} - e_i)) \\
&\overset{\text{re-index}}{=} \frac{\mathrm{u}^i_t}{q_t(\mathbf{x})} \sum_{\mathbf{y} \in \mathbb{Z}^D} \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{y})\, \Pi_{l_t}(\mathbf{x}_0 - (\mathbf{x} - e_i) + \mathbf{y}) \\
&\overset{66}{=} \mathrm{u}^i_t\, \frac{q_t(\mathbf{x} - e_i)}{q_t(\mathbf{x})}.
\end{aligned}$$

□

We can now show that the `latent-up` and `latent-down` denoisers can be expressed in terms of each other, in a non-linear and non-local way. Consequently, the `data` denoiser can be expressed in terms of a single latent denoiser, in a non-linear and non-local way.

Corollary C.4 (Relations between `latent-up`, `latent-down`, and `data` denoisers)**.** *Fix* $\mathbf{x} \in \mathbb{Z}^D, t \in (0, \tau)$ *and* $i \in [D]$*. Then,*

$$\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) = \frac{\mathrm{u}^i_t l^i_t}{\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x} - e_i, t)}, \tag{76}$$

and

$$\mathrm{m}^i_{\texttt{data}}(\mathbf{x}, t) = \mathbf{x}^i - \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) + \frac{l^i_t \mathrm{u}^i_t}{\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x} + e_i, t)} \tag{77}$$

$$= \mathbf{x}^i + \mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, t) - \frac{\mathrm{u}^i_t l^i_t}{\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x} - e_i, t)}. \tag{78}$$

Proof. For Equation (76) we use

$$\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) \overset{75}{=} \mathrm{u}^i_t \frac{q_t(\mathbf{x} - e_i)}{q_t(\mathbf{x})} \overset{74}{=} \frac{\mathrm{u}^i_t l^i_t}{\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x} - e_i, t)}.$$

Equation (77) and Equation (78) then follow from Equation (76) and the relation

$$\mathrm{m}_{\texttt{data}}(\mathbf{x}, t) = \mathbf{x} + \mathrm{m}_{\texttt{latent-down}}(\mathbf{x}, t) - \mathrm{m}_{\texttt{latent-up}}(\mathbf{x}, t).$$

□

We conclude by expressing the latent denoisers in terms of the marginal posteriors of the data given its noisy observations.

Lemma C.5. *Fix* $i \in [D]$ *and* $\mathbf{x} \in \mathbb{Z}^D$*. Then,*

$$\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, t) = \sum_{\mathbf{z}^i \in \mathbb{Z}} q_{\mathbf{x}^i_{\text{data}} | Y_t = \mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i \in \mathbb{Z}} \mathbf{y}^i\, \frac{\Pi_{\mathrm{u}^i_t}(\mathbf{y}^i) \Pi_{l^i_t}(\mathbf{y}^i + \mathbf{z}^i - \mathbf{x}^i)}{\sum_{w \in \mathbb{Z}} \Pi_{\mathrm{u}^i_t}(w) \Pi_{l^i_t}(w + \mathbf{z}^i - \mathbf{x}^i)}, \tag{79}$$

and

$$\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, t) = \sum_{\mathbf{z}^i \in \mathbb{Z}} q_{\mathbf{x}^i_{\text{data}} | Y_t = \mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i \in \mathbb{Z}} \mathbf{y}^i\, \frac{\Pi_{l^i_t}(\mathbf{y}^i) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i + \mathbf{y}^i - \mathbf{z}^i)}{\sum_{w \in \mathbb{Z}} \Pi_{l^i_t}(w) \Pi_{\mathrm{u}^i_t}(\mathbf{x}^i + w - \mathbf{z}^i)}. \tag{80}$$

Proof. We first note that Equation (70) shows that $q_{\mathrm{U}_t|Y_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}$ is a product distribution with marginals

$$q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i) = \frac{\Pi_{\mathrm{u}_t^i}(\mathbf{y}^i)\Pi_{l_t^i}(\mathbf{y}^i+\mathbf{z}^i-\mathbf{x}^i)}{\sum_{w\in\mathbb{Z}}\Pi_{\mathrm{u}_t^i}(w)\Pi_{l_t^i}(w+\mathbf{z}^i-\mathbf{x}^i)}. \tag{81}$$

By marginalization it follows that

$$\begin{aligned}
q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i}(\mathbf{y}^i) &= \sum_{\mathbf{z}^{-i}\in\mathbb{Z}^{D-1}} q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i,\mathbf{x}^{-i}_{\text{data}}=\mathbf{z}^{-i}}(\mathbf{y}^i) q_{\mathbf{x}^{-i}_{\text{data}}|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i}(\mathbf{z}^{-i}) \\
&= \sum_{\mathbf{z}^{-i}\in\mathbb{Z}^{D-1}} q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i) q_{\mathbf{x}^{-i}_{\text{data}}|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i}(\mathbf{z}^{-i}) \\
&= q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i) \underbrace{\sum_{\mathbf{z}^{-i}\in\mathbb{Z}^{D-1}} q_{\mathbf{x}^{-i}_{\text{data}}|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i}(\mathbf{z}^{-i})}_{=1} = q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}_{\text{data}}=\mathbf{z}}(\mathbf{y}^i).
\end{aligned}$$

Combining with Equation (81) we conclude

$$q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i}(\mathbf{y}^i) = \frac{\Pi_{\mathrm{u}_t^i}(\mathbf{y}^i)\Pi_{l_t^i}(\mathbf{y}^i+\mathbf{z}^i-\mathbf{x}^i)}{\sum_{w\in\mathbb{Z}}\Pi_{\mathrm{u}_t^i}(w)\Pi_{l_t^i}(w+\mathbf{z}^i-\mathbf{x}^i)}. \tag{82}$$

Hence, by Bayes' rule,

$$\begin{aligned}
\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x},t) &= \mathbb{E}[\mathrm{U}_t^i|Y_t=\mathbf{x}] = \sum_{\mathbf{y}^i\in\mathbb{Z}} \mathbf{y}^i\, q_{\mathrm{U}_t^i|Y_t=\mathbf{x}}(\mathbf{y}^i) \\
&= \sum_{\mathbf{y}^i\in\mathbb{Z}} \mathbf{y}^i \frac{q_{\mathrm{U}_t^i,Y_t}(\mathbf{y}^i,\mathbf{x})}{q_{Y_t}(\mathbf{x})} = \sum_{\mathbf{y}^i\in\mathbb{Z}} \mathbf{y}^i \frac{1}{q_{Y_t}(\mathbf{x})} \sum_{\mathbf{z}^i\in\mathbb{Z}} q_{\mathrm{U}_t^i,Y_t,\mathbf{x}^i_{\text{data}}}(\mathbf{y}^i,\mathbf{x},\mathbf{z}^i) \\
&= \sum_{\mathbf{y}^i\in\mathbb{Z}} \mathbf{y}^i \sum_{\mathbf{z}^i\in\mathbb{Z}} q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i}(\mathbf{y}^i) \frac{q_{Y_t,\mathbf{x}^i_{\text{data}}}(\mathbf{x},\mathbf{z}^i)}{q_{Y_t}(\mathbf{x})} \\
&= \sum_{\mathbf{y}^i\in\mathbb{Z}} \mathbf{y}^i \sum_{\mathbf{z}^i\in\mathbb{Z}} q_{\mathrm{U}_t^i|Y_t=\mathbf{x},\mathbf{x}^i_{\text{data}}=\mathbf{z}^i}(\mathbf{y}^i) q_{\mathbf{x}^i_{\text{data}}|Y_t=\mathbf{x}}(\mathbf{z}^i) \\
&\overset{82}{=} \sum_{\mathbf{z}^i\in\mathbb{Z}} q_{\mathbf{x}^i_{\text{data}}|Y_t=\mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i\in\mathbb{Z}} \mathbf{y}^i \frac{\Pi_{\mathrm{u}_t^i}(\mathbf{y}^i)\Pi_{l_t^i}(\mathbf{y}^i+\mathbf{z}^i-\mathbf{x}^i)}{\sum_{w\in\mathbb{Z}}\Pi_{\mathrm{u}_t^i}(w)\Pi_{l_t^i}(w+\mathbf{z}^i-\mathbf{x}^i)},
\end{aligned}$$

and similarly, using Equation (69),

$$\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x},t) = \sum_{\mathbf{z}^i\in\mathbb{Z}} q_{\mathbf{x}^i_{\text{data}}|Y_t=\mathbf{x}}(\mathbf{z}^i) \sum_{\mathbf{y}^i\in\mathbb{Z}} \mathbf{y}^i \frac{\Pi_{l_t^i}(\mathbf{y}^i)\Pi_{\mathrm{u}_t^i}(\mathbf{x}^i+\mathbf{y}^i-\mathbf{z}^i)}{\sum_{w\in\mathbb{Z}}\Pi_{l_t^i}(w)\Pi_{\mathrm{u}_t^i}(\mathbf{x}^i+w-\mathbf{z}^i)}.$$

□

C.2 Forward and backward processes

In this section we will show how to use the latent denoisers in order to define forward and backward processes with the marginals of the Poisson-Poisson perturbation (65). Consequently, we prove Theorem C.1. For $i\in[D]$ and $t\in[0,\tau]$ set

$$\mathrm{u}_0^i = 0, \quad l_0^i = 0, \qquad \dot{\mathrm{u}}_t^i := \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{u}_t^i, \qquad \dot{l}_t^i := \frac{\mathrm{d}}{\mathrm{d}t} l_t^i. \tag{83}$$

We begin by deriving the evolution equation of the marginals of the Poisson-Poisson perturbation (65).

Lemma C.6. *Let $Y_t\sim q_t$ where Y_t is as in Equation* (65)*. Then, for $\mathbf{x}\in\mathbb{Z}^D$ and $t\in[0,\tau]$,*

$$\partial_t q_t(\mathbf{x}) = \sum_{i=1}^D \dot{l}_t^i\, q_t(\mathbf{x}+e_i) + \sum_{i=1}^D \dot{\mathrm{u}}_t^i\, q_t(\mathbf{x}-e_i) - q_t(\mathbf{x}) \sum_{i=1}^D (\dot{l}_t^i + \dot{\mathrm{u}}_t^i). \tag{84}$$

Proof.

$$\partial_t q_t(\mathbf{x}) \overset{66}{=} \partial_t \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{y})\, \Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0)$$

$$= \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0) \partial_t \Pi_{l_t}(\mathbf{y}) + \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{y}) \partial_t \Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0)$$

$$\overset{16}{=} \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0) \left[\sum_{i=1}^{D} \dot{l}_t^i \left(\Pi_{l_t}(\mathbf{y}-e_i) - \Pi_{l_t}(\mathbf{y}) \right) \right]$$

$$+ \sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{y}) \left[\sum_{i=1}^{D} \dot{\mathrm{u}}_t^i \left(\Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0-e_i) - \Pi_{\mathrm{u}_t}(\mathbf{x}+\mathbf{y}-\mathbf{x}_0) \right) \right]$$

$$= \sum_{i=1}^{D} \dot{l}_t^i \left(\sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{\mathrm{u}_t}((\mathbf{x}+e_i)+(\mathbf{y}-e_i)-\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{y}-e_i) \right) - q_t(\mathbf{x}) \sum_{i=1}^{D} \dot{l}_t^i$$

$$+ \sum_{i=1}^{D} \dot{\mathrm{u}}_t^i \left(\sum_{\mathbf{x}_0 \in \mathbb{Z}^D} \sum_{\mathbf{y} \in \mathbb{Z}^D} p_{\text{data}}(\mathbf{x}_0)\, \Pi_{l_t}(\mathbf{y})\, \Pi_{\mathrm{u}_t}((\mathbf{x}-e_i)+\mathbf{y}-\mathbf{x}_0) \right) - q_t(\mathbf{x}) \sum_{i=1}^{D} \dot{\mathrm{u}}_t^i$$

$$= \sum_{i=1}^{D} \dot{l}_t^i\, q_t(\mathbf{x}+e_i) + \sum_{i=1}^{D} \dot{\mathrm{u}}_t^i\, q_t(\mathbf{x}-e_i) - q_t(\mathbf{x}) \sum_{i=1}^{D} (\dot{l}_t^i + \dot{\mathrm{u}}_t^i).$$

□

Next we will show how to use the denoisers (61) to define a CTMC whose marginals are the same as the Poisson-Poisson perturbation (65).

Proposition C.7. *Let $(\overrightarrow{\mathbf{Y}_t})_{t\in[0,\tau]}$ be the CTMC with rate matrices*

$$\overrightarrow{Q}_{\mathbf{xy}}(t) = \begin{cases} \dot{\mathrm{u}}_t^i & \text{if} \quad \mathbf{y} = \mathbf{x}+e_i, \qquad i \in [D] \\ \dot{l}_t^i & \text{if} \quad \mathbf{y} = \mathbf{x}-e_i, \qquad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x}\pm e_1, \ldots, \mathbf{x}\pm e_D\} \\ -\sum_{\mathbf{y} \in \{\mathbf{x}\pm e_1, \ldots, \mathbf{x}\pm e_D\}} \overrightarrow{Q}_{\mathbf{xy}}(t) & \text{if} \quad \mathbf{y} = \mathbf{x} \end{cases}, \tag{85}$$

and $\overrightarrow{\mathbf{Y}_0} \sim p_{\text{data}}$. Then,

$$\overrightarrow{\mathbf{Y}_t} \sim q_t \qquad \text{for all} \qquad t \in [0,\tau].$$

Proof. It suffices to show that $(q_t)_{t\in[0,\tau]}$ satisfies the Kolmogorov equation

$$\partial_t q_t(\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^D} q_t(\mathbf{y}) \overrightarrow{Q}_{\mathbf{yx}}(t). \tag{86}$$

This holds true by Equation (84). □

We can now derive the rates of the time-reversal of $(\overrightarrow{\mathbf{Y}_t})_{t\in[0,\tau]}$.

Corollary C.8. *With $s = \tau - t$, the time reversal $(\overleftarrow{\mathbf{Y}_s})_{s\in[0,\tau]}$ of $(\overrightarrow{\mathbf{Y}_t})_{t\in[0,\tau]}$ has, for $0 < s < \tau$, rate matrices*

$$\overleftarrow{Q}_{\mathbf{xy}}(s) = \begin{cases} \frac{\dot{\mathrm{u}}^i_{\tau-s}}{\mathrm{u}^i_{\tau-s}} \mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x}, \tau-s) & \text{if} \quad \mathbf{y} = \mathbf{x}-e_i, \qquad i \in [D] \\ \frac{\dot{l}^i_{\tau-s}}{l^i_{\tau-s}} \mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x}, \tau-s) & \text{if} \quad \mathbf{y} = \mathbf{x}+e_i, \qquad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x}\pm e_1, \ldots, \mathbf{x}\pm e_D\} \\ -\sum_{\mathbf{y} \in \{\mathbf{x}\pm e_1, \ldots, \mathbf{x}\pm e_D\}} \overleftarrow{Q}_{\mathbf{xy}}(s) & \text{if} \quad \mathbf{y} = \mathbf{x} \end{cases}. \tag{87}$$

Proof. By the time-reversal formula

$$\overleftarrow{Q}_{\mathbf{xy}}(s) = \frac{q_{\tau-s}(\mathbf{y})}{q_{\tau-s}(\mathbf{x})}\,\overrightarrow{Q}_{\mathbf{yx}}(\tau - s),$$

so

$$\overleftarrow{Q}_{\mathbf{x},\mathbf{x}-e_i}(s) = \frac{q_{\tau-s}(\mathbf{x}-e_i)}{q_{\tau-s}(\mathbf{x})}\overrightarrow{Q}_{\mathbf{x}-e_i,\mathbf{x}}(\tau-s) \overset{75}{=} \frac{\dot{\mathrm{u}}^i_{\tau-s}}{\mathrm{u}^i_{\tau-s}}\mathrm{m}^i_{\texttt{latent-up}}(\mathbf{x},\tau-s),$$

and

$$\overleftarrow{Q}_{\mathbf{x},\mathbf{x}+e_i}(s) = \frac{q_{\tau-s}(\mathbf{x}+e_i)}{q_{\tau-s}(\mathbf{x})}\overrightarrow{Q}_{\mathbf{x}+e_i,\mathbf{x}}(\tau-s) \overset{74}{=} \frac{\dot{l}^i_{\tau-s}}{l^i_{\tau-s}}\,\mathrm{m}^i_{\texttt{latent-down}}(\mathbf{x},\tau-s).$$

□

C.3 Proof of Theorem C.1

The time-reversed chain in Corollary C.8, indexed by sampling time s, starts from q_τ and has marginals $q_{\tau-s}$. Its rates are those in Equation (62) under the notation change Equation (64).

D Jumping only up or only down

In this section we work out special cases of the Binomial-Poisson process and Poisson-Poisson process where the jumps are uni-directional. The displays below give the off-diagonal rates; the diagonal entries are minus the sum of the off-diagonal entries in the same row.

D.1 Jumping up on $\mathbb{N}^D$ or $\mathbb{Z}^D$

Binomial-Poisson. Take $l^i_t = 1$ for all $t \in [0,\tau]$ and $i \in [D]$. Then

$$X_t = \mathbf{x}_{\text{data}} + \mathrm{N}_t, \qquad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t),$$

so

$$\mathrm{m}_{\texttt{latent}}(\mathbf{x},t) = \mathbb{E}[\mathbf{x}_{\text{data}}|\mathbf{x}_{\text{data}} + \mathrm{N}_t = \mathbf{x}] = \mathrm{m}_{\texttt{data}}(\mathbf{x},t). \tag{88}$$

As $\dot{l}^i_t = 0$ for all $t \in [0,\tau]$ and $i \in [D]$, Equation (24) reads

$$Q_{\mathbf{xy}}(t) = \begin{cases} \frac{\dot{\mathrm{u}}^i_{\tau-t}}{\mathrm{u}^i_{\tau-t}}(\mathbf{x}^i - \mathrm{m}^i_{\texttt{data}}(\mathbf{x},\tau-t)) & \text{if} \quad \mathbf{y} = \mathbf{x} - e_i, \quad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} = \mathbf{x} + e_i, \quad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\} \end{cases}. \tag{89}$$

Poisson-Poisson. Take $l^i_t = 0$ for all $t \in [0,\tau]$ and $i \in [D]$. Then

$$Y_t = \mathbf{x}_{\text{data}} + \mathrm{U}_t, \qquad \mathrm{U}_t \sim \mathrm{Poisson}(\mathrm{u}_t),$$

so

$$\mathrm{m}_{\texttt{latent-down}}(\mathbf{x},t) = 0, \qquad \mathrm{m}_{\texttt{latent-up}}(\mathbf{x},t) = \mathbf{x} - \mathbb{E}[\mathbf{x}_{\text{data}}|\mathbf{x}_{\text{data}} + \mathrm{U}_t = \mathbf{x}] = \mathbf{x} - \mathrm{m}_{\texttt{data}}(\mathbf{x},t).$$

Equation (62) is then the same as Equation (89) so the "only jump up" version of the Binomial-Poisson process and the Poisson-Poisson process is the same.

D.2 Jumping down on $\mathbb{N}^D$

Take $\mathrm{u}^i_t = 0$ for all $t \in [0,\tau]$ and $i \in [D]$. Then

$$X_t = \mathrm{B}_t, \qquad \mathrm{B}_t \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}, l_t),$$

so

$$\mathrm{m}_{\texttt{data}}(\mathbf{x},t) = \mathbb{E}[\mathbf{x}_{\text{data}}|\mathrm{B}_t = \mathbf{x}], \qquad \mathrm{m}_{\texttt{latent}}(\mathbf{x},t) = \mathbf{x},$$

and hence Equation (24) reads

$$Q_{\mathbf{xy}}(t) = \begin{cases} 0 & \text{if} \quad \mathbf{y} = \mathbf{x} - e_i, \quad i \in [D] \\ -\frac{\dot{l}^i_{\tau-t}}{1-l^i_{\tau-t}}(\mathrm{m}^i_{\texttt{data}}(\mathbf{x},\tau-t) - \mathbf{x}^i) & \text{if} \quad \mathbf{y} = \mathbf{x} + e_i, \quad i \in [D] \\ 0 & \text{if} \quad \mathbf{y} \notin \{\mathbf{x}, \mathbf{x} \pm e_1, \ldots, \mathbf{x} \pm e_D\} \end{cases}. \tag{90}$$

Equation (90) is exactly the Binomial Flow of Shenfeld et al. (2026).

E EDM SCALING

In this section we derive centered affine EDM scaling for both the Binomial-Poisson and Poisson-Poisson perturbations. The regression target may itself depend on time: let A_σ be the target and Y_σ the observed noisy input, with joint distribution

$$\pi_\sigma(a, y) = \Pr(A_\sigma = a, Y_\sigma = y).$$

This includes a fixed data target $\mathbf{x}_{\text{data}}$ and time-dependent latent targets $(\mathrm{B}_t, \mathrm{N}_t, \mathrm{L}_t)$. For the purpose of computing scalar preconditioning coefficients, we assume that the coordinates $i \in \{1, \dots, D\}$ of A_σ and Y_σ are i.i.d., with coordinatewise moments

$$\begin{aligned}\mathbb{E}[A^i_\sigma] = \mu_A(\sigma), \quad \mathrm{Var}[A^i_\sigma] = v_A(\sigma),\\ \mathbb{E}[Y^i_\sigma] = \mu_Y(\sigma), \quad \mathrm{Var}[Y^i_\sigma] = v_Y(\sigma), \quad \mathrm{Cov}(A^i_\sigma, Y^i_\sigma) = \mathrm{cov}_{A,Y}(\sigma).\end{aligned}$$

We parametrize the denoiser as

$$\mathrm{m}_\theta(\mathbf{y}, \sigma) = c_{\text{skip}}(\sigma)\mathbf{y} + c_{\text{bias}}(\sigma) + c_{\text{out}}(\sigma)F_\theta(c_{\text{in}}(\sigma)\mathbf{y} + s_{\text{in}}(\sigma), \sigma), \tag{91}$$

where F_θ is a neural network.

We first normalize the network input. We begin with the variance.

Claim E.1 (c_{in})**.** *If*

$$c_{\text{in}}(\sigma) = \frac{1}{\sqrt{v_Y(\sigma)}}, \tag{92}$$

then $\mathrm{Var}[c_{\text{in}}(\sigma)Y^i_\sigma + s_{\text{in}}(\sigma)] = 1$.

Proof. Immediate. □

Next s_{in} is chosen to center the input.

Claim E.2 (s_{in})**.** *If*

$$s_{\text{in}}(\sigma) = -\frac{\mu_Y(\sigma)}{\sqrt{v_Y(\sigma)}}, \tag{93}$$

and $c_{\text{in}}(\sigma)$ *is as in Equation* (92)*, then* $\mathbb{E}[c_{\text{in}}(\sigma)Y^i_\sigma + s_{\text{in}}(\sigma)] = 0$.

Proof. Immediate. □

While the coefficients c_{in} and s_{in} depend only on the statistics of the noisy observation Y_σ, the coefficients $c_{\text{bias}}, c_{\text{skip}}, c_{\text{out}}$ depend also on the target A_σ. We begin by matching the mean of the regression residual to the mean of the target using c_{bias}. Note that at this point c_{bias} is expressed in terms of c_{skip}, which will be determined later.

Claim E.3 (c_{bias})**.** *If*

$$c_{\text{bias}}(\sigma) = \mu_A(\sigma) - c_{\text{skip}}(\sigma)\mu_Y(\sigma), \tag{94}$$

then $\mathbb{E}[c_{\text{skip}}(\sigma)Y^i_\sigma + c_{\text{bias}}(\sigma)] = \mu_A(\sigma)$.

Proof. Immediate. □

We now determine the c_{skip} and c_{out}. With Equation (91), the mean-square error objective reads

$$\begin{aligned}&\int \sum_{a,y} w^2(\sigma)\,|a - \mathrm{m}_\theta(y, \sigma)|^2\,\pi_\sigma(a, y)\,\mathrm{d}\sigma\\ &= \int \sum_{a,y} w^2(\sigma)c^2_{\text{out}}(\sigma)\,|F_\theta(c_{\text{in}}(\sigma)y + s_{\text{in}}(\sigma), \sigma) - F_{\text{target}}(a, y, \sigma)|^2\,\pi_\sigma(a, y)\,\mathrm{d}\sigma,\end{aligned} \tag{95}$$

where the sum is over the support of π_σ, and

$$F_{\text{target}}(a, y, \sigma) := \frac{a - c_{\text{skip}}(\sigma)y - c_{\text{bias}}(\sigma)}{c_{\text{out}}(\sigma)}. \tag{96}$$

By the choice of c_{bias} according to Equation (94), each coordinate of this target has zero mean. We choose c_{out} so that each coordinate also has unit variance.

Claim E.4 (Relation between c_{out} and c_{skip})**.** *If*

$$c_{\text{out}}^2(\sigma) = v_A(\sigma) + c_{\text{skip}}^2(\sigma) v_Y(\sigma) - 2c_{\text{skip}}(\sigma)\text{cov}_{A,Y}(\sigma), \tag{97}$$

then

$$\text{Var}[F_{\text{target}}^i(A_\sigma, Y_\sigma, \sigma)] = 1.$$

Proof. Subtracting the deterministic bias does not change the variance, so

$$\text{Var}[F_{\text{target}}^i(A_\sigma, Y_\sigma, \sigma)] = \frac{v_A(\sigma) + c_{\text{skip}}^2(\sigma) v_Y(\sigma) - 2c_{\text{skip}}(\sigma)\text{cov}_{A,Y}(\sigma)}{c_{\text{out}}^2(\sigma)}.$$

□

To summarize, c_{in}, s_{in} were chosen so that the input to the network has mean zero and unit variance. As for $c_{\text{bias}}, c_{\text{out}}$, for a given choice of c_{skip}, we choose $c_{\text{bias}}, c_{\text{out}}$ so that the target has mean zero and unit variance. It remains to choose c_{skip}, which we do by minimizing the residual variance.

Claim E.5 (c_{skip})**.** *The minimizer over $a \in \mathbb{R}$ of*

$$R_\sigma(a) := \text{Var}[A_\sigma^i - aY_\sigma^i] = v_A(\sigma) + a^2 v_Y(\sigma) - 2a\text{cov}_{A,Y}(\sigma)$$

is

$$c_{\text{skip}}(\sigma) := \frac{\text{cov}_{A,Y}(\sigma)}{v_Y(\sigma)}. \tag{98}$$

Proof. The quadratic $R_\sigma(a)$ has derivative $2av_Y(\sigma) - 2\text{cov}_{A,Y}(\sigma)$, which vanishes at Equation (98). □

Claim E.6 (Resulting c_{out})**.** *If $c_{\text{skip}}(\sigma)$ is as in Equation* (98) *then*

$$C(\sigma) := \min_a R_\sigma(a) = v_A(\sigma) - \frac{\text{cov}_{A,Y}^2(\sigma)}{v_Y(\sigma)} \qquad \text{and} \qquad c_{\text{out}}(\sigma) = \sqrt{C(\sigma)}. \tag{99}$$

Proof. Substitute Equation (98) into Equation (97). □

Finally, we choose the effective squared-error weight $w^2(\sigma)$ so that the expected loss at initialization $F_\theta = 0$ is constant in σ.

Claim E.7 (w)**.** *If*

$$w^2(\sigma) = \frac{1}{C(\sigma)} = \frac{1}{c_{\text{out}}^2(\sigma)}, \tag{100}$$

then, for all σ,

$$\sum_{a,y} w^2(\sigma) \left| a - c_{\text{skip}}(\sigma) y - c_{\text{bias}}(\sigma) \right|^2 \pi_\sigma(a, y) = D.$$

Proof. Each residual coordinate has zero mean and variance $C(\sigma)$, so its second moment is $C(\sigma)$. Summing over coordinates and multiplying by $1/C(\sigma)$ gives D. □

We summarize the EDM scalings in Table 4.

E.1 EDM SCALING FOR JUD

In this section we instantiate the EDM scaling for JUD, both for the Binomial-Poisson perturbation and for the Poisson-Poisson perturbation. This requires us to compute the statistics for the observations X_t and Y_t, as well as the targets $\mathbf{x}_{\text{data}}, \mathrm{B}_t, \mathrm{N}_t, \mathrm{L}_t$. The following claim will be used to this end.

Claim E.8 (Means and variances for the Binomial and Poisson perturbations)**.** *Let Z_0 be a random vector with i.i.d. coordinates with mean μ and variance v.*

Table 4: EDM scalings for target A_σ and input Y_σ.

$c_{\text{in}}(\sigma)$	$\frac{1}{\sqrt{v_Y(\sigma)}}$
$s_{\text{in}}(\sigma)$	$-\frac{\mu_Y(\sigma)}{\sqrt{v_Y(\sigma)}}$
$c_{\text{skip}}(\sigma)$	$\frac{\text{cov}_{A,Y}(\sigma)}{v_Y(\sigma)}$
$c_{\text{bias}}(\sigma)$	$\mu_A(\sigma) - c_{\text{skip}}(\sigma)\mu_Y(\sigma)$
$C(\sigma)$	$v_A(\sigma) - \frac{\text{cov}^2_{A,Y}(\sigma)}{v_Y(\sigma)}$
$c_{\text{out}}(\sigma)$	$\sqrt{C(\sigma)}$
$w^2(\sigma)$	$\frac{1}{C(\sigma)}$

1. *Let* $Z_\sigma = \mathrm{B}_\sigma$ *with* $\mathrm{B}_\sigma \sim \mathrm{Binomial}(Z_0, l_\sigma)$. *Then,*

$$\mathbb{E}[Z_\sigma] = l_\sigma \mu, \quad \mathrm{Var}[Z_\sigma] = \mu l_\sigma (1 - l_\sigma) + l_\sigma^2 v, \quad \mathrm{Cov}(Z_0, Z_\sigma) = l_\sigma v. \tag{101}$$

2. *Let* $Z_\sigma = Z_0 \pm \mathrm{N}_\sigma$ *with* $\mathrm{N}_\sigma \sim \mathrm{Poisson}(\mathrm{u}_\sigma)$ *where* N_σ *is independent of* Z_0. *Then,*

$$\mathbb{E}[Z_\sigma] = \mu \pm \mathrm{u}_\sigma, \quad \mathrm{Var}[Z_\sigma] = v + \mathrm{u}_\sigma, \quad \mathrm{Cov}(Z_0, Z_\sigma) = v. \tag{102}$$

3. *Let* $Z_\sigma = \mathrm{B}_\sigma + \mathrm{N}_\sigma$ *with* $\mathrm{B}_\sigma \sim \mathrm{Binomial}(Z_0, l_\sigma)$, $\mathrm{N}_\sigma \sim \mathrm{Poisson}(\mathrm{u}_\sigma)$ *where* N_σ *is independent of* (Z_0, B_σ). *Then,*

$$\mathbb{E}[Z_\sigma] = \mu\, l_\sigma + \mathrm{u}_\sigma, \quad \mathrm{Var}[Z_\sigma] = \mu l_\sigma (1 - l_\sigma) + l_\sigma^2 v + \mathrm{u}_\sigma, \quad \mathrm{Cov}(Z_0, Z_\sigma) = l_\sigma v. \tag{103}$$

Proof.

1. Using the mean and variance of the binomial distribution,

$$\mathbb{E}[Z_\sigma] = \mathbb{E}[\mathbb{E}[Z_\sigma | Z_0]] = \mathbb{E}[Z_0\, l_\sigma] = l_\sigma \mu,$$

$$\begin{aligned}\mathrm{Var}[Z_\sigma] &= \mathbb{E}[\mathrm{Var}[Z_\sigma | Z_0]] + \mathrm{Var}(\mathbb{E}[Z_\sigma | Z_0]) = \mathbb{E}[Z_0\, l_\sigma (1 - l_\sigma)] + \mathrm{Var}(Z_0\, l_\sigma) \\ &= \mu l_\sigma (1 - l_\sigma) + l_\sigma^2 v,\end{aligned}$$

and

$$\begin{aligned}\mathrm{Cov}(Z_0, Z_\sigma) &= \mathbb{E}[Z_0 Z_\sigma] - \mathbb{E}[Z_0]\mathbb{E}[Z_\sigma] = \mathbb{E}[Z_0 \mathbb{E}[Z_\sigma | Z_0]] - \mu \cdot l_\sigma \mu \\ &= l_\sigma \mathbb{E}[Z_0^2] - \mu^2 l_\sigma = l_\sigma (v + \mu^2) - \mu^2 l_\sigma = l_\sigma v.\end{aligned}$$

2. Using the mean and variance of the Poisson distribution, and the independence of N_σ from Z_0,

$$\mathbb{E}[Z_\sigma] = \mathbb{E}[\mathbb{E}[Z_\sigma | Z_0]] = \mathbb{E}[\mathbb{E}[Z_0 \pm \mathrm{N}_\sigma | Z_0]] = \mu \pm \mathrm{u}_\sigma,$$

$$\begin{aligned}\mathrm{Var}[Z_\sigma] &= \mathbb{E}[\mathrm{Var}[Z_\sigma | Z_0]] + \mathrm{Var}(\mathbb{E}[Z_\sigma | Z_0]) = \mathbb{E}[\mathrm{Var}[Z_0 \pm \mathrm{N}_\sigma | Z_0]] + \mathrm{Var}(\mathbb{E}[Z_0 \pm \mathrm{N}_\sigma | Z_0]) \\ &= \mathbb{E}[\mathrm{Var}[\pm \mathrm{N}_\sigma]] + \mathrm{Var}(Z_0 \pm \mathrm{u}_\sigma) = \mathrm{u}_\sigma + v,\end{aligned}$$

and

$$\begin{aligned}\mathrm{Cov}(Z_0, Z_\sigma) &= \mathbb{E}[Z_0 Z_\sigma] - \mathbb{E}[Z_0]\mathbb{E}[Z_\sigma] = \mathbb{E}[Z_0(Z_0 \pm \mathrm{N}_\sigma)] - \mathbb{E}[Z_0]\mathbb{E}[Z_0 \pm \mathrm{N}_\sigma] \\ &= \mathbb{E}[Z_0^2] \pm \mu \mathrm{u}_\sigma - \mu(\mu \pm \mathrm{u}_\sigma) = (v + \mu^2) - \mu^2 = v.\end{aligned}$$

3. Using the means and variances of the Binomial and Poisson distributions, and the independence of N_σ from (Z_0, B_σ),

$$\begin{aligned}\mathbb{E}[Z_\sigma] &= \mathbb{E}[\mathbb{E}[Z_\sigma|Z_0]] = \mathbb{E}[\mathbb{E}[\mathrm{B}_\sigma + \mathrm{N}_\sigma|Z_0]] = \mathbb{E}[\mathbb{E}[\mathrm{B}_\sigma|Z_0] + \mathrm{N}_\sigma] \\ &= \mathbb{E}[Z_0\, l_\sigma] + \mathrm{u}_\sigma = \mu\, l_\sigma + \mathrm{u}_\sigma,\end{aligned}$$

$$\begin{aligned}\mathrm{Var}[Z_\sigma] &= \mathbb{E}[\mathrm{Var}[Z_\sigma|Z_0]] + \mathrm{Var}(\mathbb{E}[Z_\sigma|Z_0]) = \mathbb{E}[\mathrm{Var}[\mathrm{B}_\sigma + \mathrm{N}_\sigma|Z_0]] + \mathrm{Var}(\mathbb{E}[\mathrm{B}_\sigma + \mathrm{N}_\sigma|Z_0]) \\ &= \mathbb{E}[\mathrm{Var}[\mathrm{B}_\sigma|Z_0]] + \mathrm{Var}[\mathrm{N}_\sigma] + \mathrm{Var}(\mathbb{E}[\mathrm{B}_\sigma|Z_0] + \mathbb{E}[\mathrm{N}_\sigma]) \\ &= \mathbb{E}[Z_0\, l_\sigma(1-l_\sigma)] + \mathrm{u}_\sigma + \mathrm{Var}(\mathbb{E}[\mathrm{B}_\sigma|Z_0]) = \mu l_\sigma(1-l_\sigma) + \mathrm{u}_\sigma + \mathrm{Var}(Z_0\, l_\sigma) \\ &= \mu l_\sigma(1-l_\sigma) + \mathrm{u}_\sigma + l_\sigma^2 v,\end{aligned}$$

and

$$\begin{aligned}\mathrm{Cov}(Z_0, Z_\sigma) &= \mathbb{E}[Z_0 Z_\sigma] - \mathbb{E}[Z_0]\mathbb{E}[Z_\sigma] = \mathbb{E}[Z_0\mathbb{E}[\mathrm{B}_\sigma + \mathrm{N}_\sigma|Z_0]] - \mu\mathbb{E}[\mathrm{B}_\sigma + \mathrm{N}_\sigma] \\ &= \mathbb{E}[Z_0(\mathbb{E}[\mathrm{B}_\sigma|Z_0] + \mathbb{E}[\mathrm{N}_\sigma])] - \mu(\mu l_\sigma + \mathrm{u}_\sigma) \\ &= \mathbb{E}[Z_0(Z_0 l_\sigma + \mathrm{u}_\sigma)] - \mu(\mu l_\sigma + \mathrm{u}_\sigma) = l_\sigma\mathbb{E}[Z_0^2] + \mu\mathrm{u}_\sigma - \mu^2 l_\sigma - \mathrm{u}_\sigma\mu \\ &= l_\sigma(v + \mu^2) - \mu^2 l_\sigma = l_\sigma v.\end{aligned}$$

□

Binomial-Poisson moments. Let the clean data coordinates have mean μ and variance v, and write

$$X_t = \mathrm{B}_t + \mathrm{N}_t, \qquad \mathrm{B}_t \mid \mathbf{x}_{\text{data}} \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}, l_t), \qquad \mathrm{N}_t \sim \mathrm{Poisson}(\mathrm{u}_t), \qquad \mathrm{L}_t := \mathbf{x}_{\text{data}} - \mathrm{B}_t,$$

where N_t is independent of $(\mathbf{x}_{\text{data}}, \mathrm{B}_t)$. Here and below the schedules are scalar, or the formulas are applied coordinatewise. Define

$$V_B(t) := \mu l_t(1-l_t) + l_t^2 v, \qquad V_L(t) := \mu l_t(1-l_t) + (1-l_t)^2 v.$$

We apply the generic EDM formulas with $\sigma = t$ and $Y_\sigma = X_t$. By Claim E.8(3), the observation moments are

$$\mu_Y(t) = l_t\mu + \mathrm{u}_t, \qquad v_Y(t) = V_B(t) + \mathrm{u}_t. \tag{104}$$

The target-specific moments are listed in Table 5, and follow from Claim E.8, together with the identities

$$\mathrm{Cov}(\mathrm{L}_t^i, X_t^i) = \mathrm{Cov}(\mathbf{x}_{\text{data}}^i - \mathrm{B}_t^i, \mathrm{B}_t^i + \mathrm{N}_t^i) = l_t v - V_B(t) = l_t(1-l_t)(v-\mu),$$

and

$$\mathbb{E}[\mathrm{N}_t \mid X_t] = X_t - \mathbb{E}[\mathrm{B}_t \mid X_t], \qquad \mathbb{E}[\mathrm{L}_t \mid X_t] = \mathbb{E}[\mathbf{x}_{\text{data}} \mid X_t] - \mathbb{E}[\mathrm{B}_t \mid X_t].$$

Table 5: Moments of the regression targets for the Binomial-Poisson perturbation $X_t = \mathrm{B}_t + \mathrm{N}_t$.

Target A_t	$\mu_A(t)$	$v_A(t)$	$\mathrm{cov}_{A,Y}(t) = \mathrm{Cov}(A_t^i, Y_t^i)$
B_t	$l_t\mu$	$V_B(t)$	$V_B(t)$
$\mathbf{x}_{\text{data}}$	μ	v	$l_t v$
N_t	u_t	u_t	u_t
L_t	$(1-l_t)\mu$	$V_L(t)$	$l_t(1-l_t)(v-\mu)$

Poisson-Poisson moments. Let the clean data coordinates have mean μ and variance v, and write

$$Y_t = \mathbf{x}_{\text{data}} - \mathrm{D}_t + \mathrm{U}_t, \qquad \mathrm{D}_t \sim \mathrm{Poisson}(l_t), \qquad \mathrm{U}_t \sim \mathrm{Poisson}(\mathrm{u}_t),$$

with $\mathbf{x}_{\text{data}}, \mathrm{D}_t, \mathrm{U}_t$ independent. By Claim E.8 the observation moments are

$$\mu_Y(t) = \mu - l_t + \mathrm{u}_t, \qquad v_Y(t) = v + l_t + \mathrm{u}_t. \tag{105}$$

The moments of the targets follow from Claim E.8 and can be found in Table 6.

Table 6: Moments of the regression targets for the Poisson-Poisson perturbation $Y_t = \mathbf{x}_{\text{data}} - \mathrm{D}_t + \mathrm{U}_t$.

Target A_t	$\mu_A(t)$	$v_A(t)$	$\mathrm{cov}_{A,Y}(t) = \mathrm{Cov}(A_t^i, Y_t^i)$
U_t	u_t	u_t	u_t
D_t	l_t	l_t	$-l_t$

F Synthetic implementation details

F.1 Data

We draw $N = 20{,}000$ i.i.d. samples from each of the following one-dimensional target distributions for the synthetic datasets with support S:

1. Poisson ($\lambda = 5$, $S = 40$): $p(x) \propto \lambda^x e^{-\lambda}/x!$
2. Poisson mixture ($S = 140$): $p(x) \propto \sum_i w_i \lambda_i^x e^{-\lambda_i}/x!$, $w = (0.1, 0.9)$, $\lambda = (1, 100)$
3. Zero-inflated Poisson ($S = 50$): $p(0) = \pi_0 + (1 - \pi_0)e^{-\lambda}$, $p(x) = (1 - \pi_0)\lambda^x e^{-\lambda}/x!$ for $x > 0$; $\pi_0 = 0.7$, $\lambda = 5$
4. Negative binomial mixture ($S = 150$): $p(x) \propto \sum_i w_i \binom{x+r_i-1}{x} p_i^{r_i}(1 - p_i)^x$, $w = (0.8, 0.2)$, $r = (1, 10)$, $p = (0.9, 0.1)$
5. Beta–negative-binomial ($S = 100$): $p \sim \mathrm{Beta}(1.5, 1.5)$, $X|p \sim \mathrm{NegBinom}(r{=}5, p)$, marginalized by quadrature
6. Zipf ($S = 50$): $p(x) \propto x^{-\alpha}$, $\alpha = 1.7$
7. Yule–Simon ($S = 50$): $p(x) \propto \rho B(x, \rho + 1)$, $\rho = 2$
8. Correlated Beta-Binomial (dim 1, $S = 65$): $p \sim \mathrm{Beta}(2, 5)$, $X|p \sim \mathrm{Binomial}(64, p)$, giving the Beta-Binomial marginal $\binom{n}{x} B(x + \alpha, n - x + \beta)/B(\alpha, \beta)$

F.2 Training

Model. We evaluate all models using a common backbone model based on a residual MLP. A scalar input $x \in \mathbb{R}$ is first rescaled using EDM preconditioning by $z = c_{\text{in}}(t)x + s_{\text{in}}(t)$ and linearly projected to a hidden width $d_{\text{model}} = 128$. The diffusion time $t \in [T_0, T]$ is embedded via a sinusoidal timestep embedding with base frequency 10^4, and time-scale factor 100 to an embedding dimension $d_{\text{temb}} = 128$. This is passed through a two-layer MLP (ReLU, hidden width 256) to produce a $4d_{\text{temb}}$-dimensional conditioning vector. The main layers consist of 3 residual blocks with 256 hidden units followed by FiLM modulation. A method-specific head is applied to the final hidden state $h \in \mathbb{R}^{128}$: a regression head for denoisers or a soft-plus for logit predictions. For the models that require two latent denoisers or probabilities to predict birth and death rates, we use a model with two heads that share the same hidden structure.

Training. We use the Adam optimizer with the learning rate 2×10^{-4}, no weight decay, batch size 256, gradient-norm clipped to 1.0, EMA decay 0.999 and train for 300 epochs. We compute the mean-squared error loss for the Binomial-Poisson model between the model's prediction and the appropriate conditional-mean target, averaged over a batch of times $t \sim \mathrm{Unif}(T_0, T)$ and one forward-noised sample per example. For the count-fm and the denoising score-entropy loss, the loss is based on the Kullback-Leibler divergence.

F.3 Sampling

To evaluate the model, we draw 1000 samples per method using 2000-step reverse-time Euler discretization between $T = 1$ and $T_0 = 10^{-4}$. We observe that similar results were obtained with tau-leaping. Each process is initialized from its respective reference distribution that is described in Table 7. The accuracy is reported as the Wasserstein-1 distance to a fresh batch of 1000 held-out

Table 7: Reference distributions used by different methods.

Method	Reference distribution
Binomial-Poisson	$\mathrm{Poisson}(20)$
Binomial (Shenfeld et al., 2026)	δ_0
Poisson	$\mathrm{Poisson}(196)$
Count Bridge (Fishman et al., 2026)	$\mathrm{Skellam}(20, 20)$
Count Flow (Wei & Pearson, 2026)	$\mathrm{Unif}\{0, \dots, 149\}$
SEDD (Lou et al., 2024)	$\mathrm{Unif}\{0, \dots, 149\}$

samples from the true distribution. We averaged the result across 5 trials: we report the mean and standard deviation of the results.

F.4 Details on Figure 1

We use the Negative Binomial mixture (Section F.1) as the same target distribution for all methods. The network we use across models is the same as in Section F.2. The Binomial-Poisson model is trained using Algorithm 1 and the Poisson model trains the data denoiser (Section D). The rest of the methods are trained per the published implementations. Sampling is done as in Section F.3. The reference distributions are described in Table 7.

G CIFAR-10 implementation details

G.1 Training

Architecture and optimization. JUD and our EDM reproduction use DDPM++ with base width 128, channel multipliers $(2, 2, 2)$, and dropout 0.13. Both use Adam with batch size 512 and learning rate 0.001, linearly warmed up over 10 million images and then held constant. The training budget is 200 million images. EMA uses a target half-life of $500{,}000$ images with ramp-up.

Data and corruption. We use the 50,000 CIFAR-10 training images without class labels. Image augmentation is applied before clipping and rounding the pixel values to integers in $[0, 255]$. For each image, we draw one perturbation time $t \in [\varepsilon, 1-\varepsilon]$, with $\varepsilon = 10^{-5}$, and independently sample the coordinatewise perturbations

$$\mathrm{L}_t^i \mid \mathbf{x}_{\text{data}} \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}^i, t), \quad \mathrm{N}_t^i \sim \mathrm{Poisson}(\mu t), \quad X_t := \mathbf{x}_{\text{data}} - \mathrm{L}_t + \mathrm{N}_t.$$

The time distribution is specified below. The preconditioning uses the fixed scalar data moments $\mu = 120.7$ and $v = 4115.2$.

Denoisers and objective. The shared network has three input channels and six output channels, split into the predictions $(\mathrm{m}_{\texttt{latent-down}}, \mathrm{m}_{\texttt{latent-up}})$. We use the centered affine preconditioning in Table 4, with the target moments in Table 5, and provide $-\log(1 - t + \varepsilon)$ as the network's time input. Writing $C_N(t)$ and $C_L(t)$ for the two residual variances in Equation (99), the per-image loss is

$$\ell_t = \frac{1}{2D}\left(\frac{\|\mathrm{m}_{\texttt{latent-up}}(X_t, t) - \mathrm{N}_t\|_2^2}{C_N(t) + 10^{-8}} + \frac{\|\mathrm{m}_{\texttt{latent-down}}(X_t, t) - \mathrm{L}_t\|_2^2}{C_L(t) + 10^{-8}}\right), \qquad D = 3 \cdot 32^2. \tag{106}$$

Training-time distribution. As in EDM, we allocate more training examples to noise levels at which the neural denoiser captures structure beyond the affine predictor. We first train a pilot for 5 million images with t uniform on $[\varepsilon, 1-\varepsilon]$. We then evaluate its EMA denoisers on 4,096 training images, at 81 times $t_0 < \cdots < t_{80}$ uniformly spaced in $\log(t/(1-t))$ between the two endpoints.

The affine baseline is obtained by setting the learned network residual to zero in Equation (91). Let $\widehat{\ell}_{\text{aff}}(t_j)$ and $\widehat{\ell}_{\text{pilot}}(t_j)$ denote its empirical loss and that of the pilot, respectively, both evaluated using Equation (106). Define the positive improvement

$$g_j = \max\left\{\widehat{\ell}_{\text{aff}}(t_j) - \widehat{\ell}_{\text{pilot}}(t_j),\, 0\right\}, \qquad g(t) = \frac{g_j + g_{j+1}}{2} \quad \text{for } t \in [t_j, t_{j+1}).$$

For the remaining training, we sample times from a fixed mixture: 10% from a uniform proposal on $[\varepsilon, 1-\varepsilon]$ and 90% from a targeted proposal proportional to $g(t)$. The targeted proposal favors noise levels where the pilot improves more over the affine baseline.

G.2 Sampling

The CIFAR-10 model uses $\tau = 1$ and the scalar schedules, for each $i \in [D = 3 \cdot 32^2]$

$$\mathrm{d}_t^i = t, \qquad \mathrm{u}_t^i = \mu t, \qquad \mu = 120.7.$$

Sampling starts at $t = 1$ (pure noise) and decreases to $t = \varepsilon$, with $\varepsilon = 10^{-5}$.

At state $\mathbf{x}$ and time t, a single network evaluation produces the preconditioned denoiser estimates $\mathrm{m}_{\texttt{latent-down}}(\mathbf{x}, t) \approx \mathbb{E}[\mathrm{L}_t \mid X_t = \mathbf{x}]$ and $\mathrm{m}_{\texttt{latent-up}}(\mathbf{x}, t) \approx \mathbb{E}[\mathrm{N}_t \mid X_t = \mathbf{x}]$. The network's time input is transformed as $-\log(1 - t + \varepsilon)$, an adaptation of EDM's time embedding.

Sampling uses either tau-leaping or Euler as in Algorithm 3 and Algorithm 4, respectively.

G.3 Sampling ablations

Marginal-preserving rates. We augment tau-leaping with the forward-plus-reverse generator correction of Campbell et al. (2022), with unit strength. These additional rates preserve the instantaneous marginal distribution when the conditional means are exact. In Algorithm 3, replace the rates at $0 < t < 1$ by

$$\lambda_{+,\mathrm{C}}^i = \mu + \frac{2\,\mathrm{m}_{\texttt{latent-down}}^i(\mathbf{x}, t)}{t}, \qquad \lambda_{-,\mathrm{C}}^i = \frac{\mathbf{x}^i - \mathrm{m}_{\texttt{latent-up}}^i(\mathbf{x}, t)}{1 - t} + \frac{2\,\mathrm{m}_{\texttt{latent-up}}^i(\mathbf{x}, t)}{t}. \tag{107}$$

At $t = 1$, the implementation replaces $\frac{\mathbf{x}^i - \mathrm{m}_{\texttt{latent-up}}^i(\mathbf{x}, t)}{1-t}$ by $\mathbf{x}^i$. The Poisson draws and state update are unchanged. The modified rates reuse the same denoiser predictions, so there is no additional network evaluation.

Results. Table 8 varies the sampler and number of steps while keeping the trained model fixed. At 256 steps, tau-leaping obtains a substantially lower FID than Euler. At 1024 and 2048 steps, tau-leaping again obtains a slightly lower FID. Adding marginal-preserving rates worsens FID at every evaluated step count.

Table 8: CIFAR-10 FID at different numbers of sampling steps (NFE).

Sampler	256	512	1024	2048
Euler	7.07	2.22	2.15	2.09
tau-leaping	3.27	2.29	2.11	**2.07**
tau-leaping + marginal-preserving rates	5.46	2.95	2.34	2.20

H Sudoku implementation details

H.1 Training configuration

All four variants use an eight-block DiT with width 512, MLP width 2048, eight attention heads, and dropout 0.1. We use AdamW, batch size 256, peak learning rate 3×10^{-4} with warmup, gradient-norm clipping at 1, and EMA. MSE and CE share the following settings within each digit encoding.

Setting	Updates	Learning-rate schedule	Weight decay
Original digits	10,000	Warmup, then cosine decay	0.05
Shuffled digits	20,000	Warmup, then constant	0

Training times are sampled uniformly from $[0.001, 0.999]$. Digit shuffling draws a fresh independent permutation for each board presentation, applied jointly to the puzzle and solution.

H.2 Posterior parametrization

Let $\mathcal{C}$ denote the puzzle context. For each cell i, the cross-entropy network outputs nine logits whose softmax approximates the clean-digit posterior,

$$p_\theta^i(k \mid \mathbf{x}, t, \mathcal{C}) \approx \mathbb{P}(\mathbf{x}_{\text{data}}^i = k \mid X_t = \mathbf{x}, \mathcal{C}), \qquad k \in \{1, \dots, 9\}. \tag{108}$$

The conditioning includes the entire perturbed solution and puzzle. The EDM-MSE output scalings are not applied to the logits.

We use the Binomial-Poisson perturbation, $X_t^i = \mathbf{x}_{\text{data}}^i - \mathrm{L}_t^i + \mathrm{N}_t^i$, with $\mathrm{L}_t^i | \mathbf{x}_{\text{data}}^i \sim \mathrm{Binomial}(\mathbf{x}_{\text{data}}^i, \mathrm{d}_t^i)$ and independent $\mathrm{N}_t^i \sim \mathrm{Poisson}(\mathrm{u}_t^i)$, with $\mathrm{d}_t^i = t$ and $\mathrm{u}_t^i = 5t$ for all i.

Posterior averaging. As shown in Lemma B.6, we have

$$\mathrm{m}_{\texttt{latent}}^i(\mathbf{x}, t) = \sum_{k=1}^{9} p_{\mathbf{x}_{\text{data}}^i | X_t = \mathbf{x}}(k) \beta_t(\mathbf{x}^i, k) \tag{109}$$

where

$$\beta_t(\mathbf{x}^i, k) := \sum_{b=0}^{\min(\mathbf{x}^i, k)} b \, \frac{\mathcal{B}_{k, l_t^i}(b) \Pi_{\mathrm{u}_t^i}(\mathbf{x}^i - b)}{(\mathcal{B}_{k, l_t^i} * \Pi_{\mathrm{u}_t^i})(\mathbf{x}^i)}. \tag{110}$$

Hence, replacing $p_{\mathbf{x}_{\text{data}}^i | X_t = \mathbf{x}}(k)$ by the approximation $p_\theta^i(k \mid \mathbf{x}, t, \mathcal{C})$, and using the relations Equation (29)-Equation (30), we have

$$\begin{aligned} \mathrm{m}_{\texttt{latent-up}}^i(\mathbf{x}, t; \mathcal{C}) &\approx \sum_{k=1}^{9} p_\theta^i(k \mid \mathbf{x}, t, \mathcal{C}) \big[\mathbf{x}^i - \beta_t(\mathbf{x}^i, k)\big], \\ \mathrm{m}_{\texttt{latent-down}}^i(\mathbf{x}, t; \mathcal{C}) &\approx \sum_{k=1}^{9} p_\theta^i(k \mid \mathbf{x}, t, \mathcal{C}) \big[k - \beta_t(\mathbf{x}^i, k)\big]. \end{aligned} \tag{111}$$

Sampling. The reverse-time jump rates evaluated at forward noise level $0 < t \leq 1$ are

$$\lambda_+^i(\mathbf{x}, t, \mathcal{C}) = \frac{\mathrm{m}_{\texttt{latent-down}}^i(\mathbf{x}, t; \mathcal{C})}{t}, \qquad \lambda_-^i(\mathbf{x}, t, \mathcal{C}) = \frac{\mathrm{m}_{\texttt{latent-up}}^i(\mathbf{x}, t; \mathcal{C})}{t}. \tag{112}$$

These rates enter the tau-leaping update in Algorithm 3.

I Related works

Diffusion models: Continuous data Diffusion models in discrete time were first introduced in Sohl-Dickstein et al. (2015) with an evidence lower bound (ELBO) training objective. The method was then further developed in Ho et al. (2020); Song et al. (2021a), where the connection to denoising score matching was made, with additionally better parameterizations and sampling, which ultimately led to a state-of-the-art (at the time) FID score of 3.17 on CIFAR-10. The progression of diffusion models to continuous-time as reverse-time stochastic differential equations was developed in Song et al. (2021b), which also unified diffusion models with the score matching with Langevin dynamics framework (Song & Ermon, 2019). Song et al. (2021b) wrote the training objective as denoising score matching, and lowered the FID score on CIFAR-10 to 2.20. In Karras et al. (2022), the design space for training the denoiser was explored systematically, which further brought down the FID to 1.97 on CIFAR-10. The EDM framework developed in Karras et al. (2022) is regularly used in state-of-the-art results for image generation.

Discrete diffusion models: Categorical data Repeating history, the first diffusion models for categorical data were developed in discrete time (Hoogeboom et al., 2021; Austin et al., 2021), while the continuous time formulation was later developed in Campbell et al. (2022). As for continuous data, the early versions of discrete diffusion models were trained using ELBO-based objectives, specifically training the posterior distribution (not the denoiser). Score-based objectives were developed in Meng et al. (2022); Lou et al. (2024), and rate-based training was developed in Holderrieth et al. (2025).

Discrete diffusion models: Ordinal data Already at the origin of diffusion models, ordinal data appeared in the form of *binary* data (Sohl-Dickstein et al., 2015), with later developments in Bach & Saremi (2025); Bénédict et al. (2023); Pham et al. (2025). In some biological applications, e.g., gene expression counts in single-cell RNA sequencing, recent works (Zhang et al., 2026; Wang et al., 2026b; Bhattacharya et al., 2026; Wang et al., 2026a) argued that it is beneficial to develop diffusion models that act directly on the discrete data, but these works neglect the ordinal structure and use categorical discrete diffusion models.

The theory and practice for the ordinal data considered in this work, namely (subsets of) $\mathbb{N}^D$ or $\mathbb{Z}^D$, has only recently started to evolve. Let us begin with data on $\mathbb{N}^D$. One approach, developed in Chen & Zhou (2023); Montanari (2023); Bhattacharya et al. (2025), learns both a denoiser and a rate via a discrete Tweedie's formula, but using perturbation of the form $\mathbf{x}_{\text{data}} \mapsto X_t \sim \text{Poisson}(t\,\mathbf{x}_{\text{data}})$. This type of perturbation can only recover the clean data in the limit $\tau \to \infty$, in contrast to our approach which recovers $\mathbf{x}_{\text{data}}$ at a final time. In addition, this perturbation can easily overshoot the effective support of the data. Alternative approaches (Santos et al., 2023; 2025; Shenfeld et al., 2026; Cheng et al., 2026) use only Binomial noise, which suffers from being a uni-directional perturbation. Another uni-directional perturbation was developed in Campbell et al. (2023). In Winkler et al. (2024); Wei & Pearson (2026), an up-down perturbation based on Binomial noise is developed. However, this formulation does not allow for a direct training of a denoiser. In Fishman et al. (2026), the Poisson-Poisson perturbation (6) is used, but the denoiser formulation of Theorem C.1 is lacking, which leads Fishman et al. (2026) to a *distributional* training objective. To summarize, our work is the first to train a diffusion model for discrete ordinal data which allows for up-down perturbations, while training only denoisers.