MIXED POISSON DIFFUSION MODELS AND APPLICATION TO POSTERIOR SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are a powerful framework for learning complex probability distributions. While widely used diffusion models for continuous data rely on Gaussian noising processes, we instead develop diffusion models based on Poisson noise. More precisely, we introduce and study a new mathematical framework, called Mixed Poisson Diffusion Models (MPDM), with noising dynamics driven by multivariate counting processes.

Our methodology is motivated by Poisson inverse problems. The success of Gaussian diffusion models has led to their use as prior models in a growing number of posterior sampling methods for inverse problems. However, most of these approaches focus on Gaussian inverse problems, in which observations are generated through a deterministic forward operator corrupted by additive Gaussian noise. By contrast, Poisson inverse problems, where observations are discrete and Poisson distributed, remain comparatively underexplored in this literature. Building on our Poisson generative dynamics, we develop a natural framework for posterior sampling in Poisson inverse problems and derive several posterior sampling algorithms based on the proposed model. We evaluate our approach on realistic Poisson inverse problems and show that the resulting framework is well suited to this class of problems.

1 INTRODUCTION

Diffusion models have emerged as a powerful and versatile framework for learning complex probability distributions, with striking empirical success in high-dimensional generative modeling (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021). Their central principle is to define a tractable forward noising process that progressively transforms the data distribution into a simple reference distribution, and then to learn the corresponding reverse-time dynamics in order to generate new samples. In their most widely used form, diffusion models rely on Gaussian perturbations, either through discrete-time Gaussian transitions or continuous-time stochastic differential equations (SDEs). This Gaussian structure is particularly appealing from both theoretical and computational perspectives, as it leads to explicit transition kernels and efficient simulation schemes.

At the same time, recent works have demonstrated that the diffusion paradigm is not intrinsically tied to Gaussian noise. In particular, diffusion-type generative models have been extended to discrete state spaces through appropriately designed Markov chains and continuous-time jump processes (Austin et al., 2021; Campbell et al., 2022). These developments suggest a broader perspective in which the choice of the forward corruption mechanism can be adapted to the geometry or the observation model underlying the problem of interest. Yet, despite this flexibility, comparatively little attention has been devoted to diffusion models whose noising mechanism is driven by counting processes.

In this work, we introduce and study a new mathematical framework, which we call *Mixed Poisson Diffusion Models* (MPDMs), in which the forward noising dynamics are given by multivariate counting processes, and in particular mixed Poisson processes (Grandell, 1997). Rather than progressively corrupting the data by additive Gaussian perturbations, MPDMs encode the data through the random accumulation of discrete events, leading to a fundamentally different family of intermediate distributions driven by Poisson noising dynamics. The resulting framework preserves the main conceptual structure of diffusion modeling: a simulation-free noising mechanism that defines

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.

a path of probability distributions, together with a Markov process sharing the same marginals and characterized by learnable dynamics, ultimately yielding a tractable generative model.

This construction is motivated in particular by inverse problems with Poisson observations. In a Bayesian inverse problem, the goal is to recover an unknown signal of interest x from a corrupted observation y. The central object is the posterior distribution π^y of the signal, defined through Bayes' rule, up to a normalizing constant, by $\pi^y(\mathrm{d}x) \propto g(y|x)\,\pi(\mathrm{d}x)$, where π denotes the prior distribution on the unknown signal and $(y, x) \mapsto g(y|x)$ is the likelihood relating the observation to the underlying signal. Diffusion models have recently been used as expressive data-driven priors in Bayesian inverse problems and have enabled high-quality reconstructions in a broad range of imaging applications. However, sampling from π^y remains challenging because the diffusion prior is typically available only implicitly through a learned score or denoiser function. To address this difficulty, a large body of work has proposed modifying the reverse diffusion dynamics through an additional guidance term so as to generate approximate samples from the posterior distribution. Prominent examples include Diffusion Posterior Sampling (DPS) (Chung et al., 2023), Plug-and-Play Diffusion Models (PnP-DM) (Wu et al., 2024), and Decoupled Annealing Posterior Sampling (DAPS) (Zhang et al., 2025), among many others (Song et al., 2023; Kawar et al., 2022; Janati et al., 2025a); see, for instance, the review of Janati et al. (2025b).

Most existing methodologies, however, are developed or theoretically justified primarily for linear inverse problems with Gaussian observation noise. In this setting, the Gaussian structure of the likelihood is naturally compatible with the Gaussian noising process underlying standard diffusion models. In particular, for an important class of such problems, this compatibility makes it possible to exploit explicit Gaussian identities and, in some cases, to construct exact posterior transitions; see (25) and Appendix C. This favorable structure generally disappears even for simple Poisson inverse problems, where observations are discrete and the likelihood is Poisson rather than Gaussian. Developing a diffusion framework that is intrinsically compatible with Poisson observations is therefore the main applicative motivation for the methodology introduced in this work. This class of problem is common in fields such as medical imaging, e.g., Positron Emission Tomography (PET) (Bailey et al., 2005), or astronomical observation (Donath et al., 2024). These Poisson inverse problems often arise from photon-counting measurements, where observed counts can be close to or equal to zero, and one cannot reasonably default to a Gaussian noise model. Indeed, in such settings, the observation noise is inherently discrete and signal dependent, making Gaussian-based generative dynamics less naturally matched to the underlying statistical model. By building the generative process itself from Poissonian dynamics, MPDMs provide a framework that is naturally aligned with this class of observation models and, as we show, leads to new posterior-sampling methodologies for Poisson inverse problems.

Our contribution is threefold:

• In Section 2, we introduce *Mixed Poisson Diffusion Models* (MPDMs). We construct a stochastic noising dynamics based on the theory of multivariate counting processes (Protter, 2005; Daley & Vere-Jones, 2008), and notably mixed Poisson processes (Grandell, 1997; Zocher, 2005). This yields a rigorous mathematical framework for Poissonian generative modeling (Chen & Zhou, 2023; Bhattacharya et al., 2025), closely paralleling the Gaussian diffusion framework described in Section 2.2. We further establish quantitative sampling-error bounds that explicitly account for early stopping, denoiser estimation, and time-discretization errors, thereby providing a Poissonian counterpart to the error analyses commonly developed for other types of diffusion samplers.

• In Section 3, we develop a framework for *MPDM-based posterior sampling*. To the best of our knowledge, this is the first posterior sampling methodology built upon Poissonian generative dynamics. We motivate this construction through Poisson inverse problems. For simple forward operators, Theorem 3 establishes a structural connection between MPDMs and Poisson inverse problems: in contrast to Gaussian diffusion-based approaches, posterior sampling with MPDMs can be carried out without introducing substantial approximations. For more general forward operators, we adapt several established diffusion-based posterior sampling methods to the MPDM setting.

• Finally, we evaluate the proposed algorithms in Section 4. In particular we consider both synthetic and realistic Poisson inverse problems. Our experiments demonstrate the practical relevance of the framework and highlight the advantages of matching the generative noising dynamics to the Poisson structure of the observation model.

Notation. All random variable are supposed to be defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Given a topological space X equipped with its Borel σ-algebra $\mathcal{B}(\mathsf{X})$, a (continuous-time) *stochastic process* $\mathbf{X} = (X_t)_{t\geqslant 0}$ is a collection of random variables X_t valued in X, indexed by $t \in \mathbb{R}_+$. If $\mathsf{X} = \mathbb{R}^d$, we write $\mathbf{X} = (\mathbf{X}^1, \dots, \mathbf{X}^d)$, where for $i \in \{1, \dots, d\}$ and $t \in \mathbb{R}_+$, X_t^i is the i-th component of X_t. The *canonical filtration* associated to $\mathbf{X}$ is the filtration $\mathbf{F}^{\mathbf{X}} = (\mathcal{F}_t^{\mathbf{X}})_{t\geqslant 0}$, where for any $t \geqslant 0$, $\mathcal{F}_t^{\mathbf{X}} = \sigma(\mathbf{X}_s^i, 0 \leqslant s \leqslant t, i = 1, \dots, d)$. $\mathbf{X}$ is said *càdlàg* if its trajectories are almost surely right-continuous and admit left limits, *i.e.*, for almost all $\omega \in \Omega$ and any $t > 0$, $X_{t+}(\omega) \coloneqq \lim_{s\downarrow t} X_s(\omega) = X_t(\omega)$ and $X_{t-}(\omega) \coloneqq \lim_{s\uparrow t} X_s(\omega) \in \mathbb{R}^d$.

2 Mixed Poisson diffusion models

In this section, we propose our framework for Poissonian generative modeling, leveraging the theory of counting processes and specifically mixed Poisson processes (Grandell, 1997). This construction provides a continuous-time mathematical framework for Poissonian generative models. A parallel with Gaussian diffusion models is provided in Section 2.2.

2.1 Mixed Poisson processes for generative modelling

Counting processes. Unlike SDE-based diffusion models, whose trajectories are continuous, counting processes are *càdlàg*, integer-valued stochastic processes. We refer to Appendix B and (Laub et al., 2015; Protter, 2005; Zocher, 2005) for a more detailed introduction.

Definition 1. *A univariate **counting process** is a stochastic process $\mathbf{Z} = (Z_t)_{t\geqslant 0}$ whose trajectories $Z(\omega) : t \in \mathbb{R}_+ \mapsto Z_t(\omega) \in \mathbb{N}$, for almost all $\omega \in \Omega$, are càdlàg, non-decreasing and piecewise constant, with $Z_t(\omega) - Z_{t-}(\omega) \in \{0, 1\}$ for any $t > 0$ and $Z_0(\omega) = 0$. A multivariate counting process is a $\mathbb{N}^d$-valued process whose coordinates $\mathbf{Z}^i, 1 \leqslant i \leqslant d$, and norm $\|\mathbf{Z}\|_1$ are univariate counting processes.*

Alternatively, a univariate counting process can be defined through its sequence of *jump times*, *i.e.*, $Z_t = \sum_{n\geqslant 1} \mathbb{1}_{T_n \leqslant t}$ where $(T_n)_{n\geqslant 1}$ satisfies, almost surely, $0 < T_1 < T_2 < \cdots \leqslant \infty$. In the multivariate case, the condition on $\|\mathbf{Z}\|_1$ prevents simultaneous jumps: we unambiguously define the jump times $(T_n)_{n\geqslant 1}$ of $\mathbf{Z}$ as those of $\|\mathbf{Z}\|_1$. For any $n \geqslant 1$, almost surely, there is a unique pair $1 \leqslant i \leqslant d, k \geqslant 1$ such that $T_n = T_k^i$, where $(T_l^i)_{l\geqslant 1}$ are the jump times of $\mathbf{Z}^i$.

In the rest of this section, $\mathbf{Z}$ is a multivariate counting process adapted to a complete, right-continuous filtration $\mathbf{F} = (\mathcal{F}_t)_{t\geqslant 0}$, *i.e.*, for any $t \geqslant 0$, $\mathcal{F}_t^{\mathbf{Z}} \subseteq \mathcal{F}_t$ where $\mathcal{F}_t^{\mathbf{Z}} = \sigma(Z_s^i, 1 \leqslant i \leqslant d, 0 \leqslant s \leqslant t)$. $\mathbf{F}^{\mathbf{Z}} = (\mathcal{F}_t^{\mathbf{Z}})_{t\geqslant 0}$ is the canonical filtration associated to $\mathbf{Z}$. We furthermore assume that for any $t \geqslant 0$, $\mathbb{E}[\|Z_t\|_1] < \infty$. Being non-decreasing, each coordinate process $\mathbf{Z}^i, 1 \leqslant i \leqslant d$ is a $\mathbf{F}$-submartingale. As a consequence of the Doob-Meyer decomposition, we have the following result, that is a simple extension of the univariate case; see (Protter, 2005, Chapter III).

Proposition 1. *There exists a unique $\mathbf{F}$-predictable process $\mathbf{\Lambda}$ whose coordinates $\mathbf{\Lambda}^i, 1 \leqslant i \leqslant d$ are non-decreasing finite-variation processes, with $\Lambda_0^i = 0$ a.s., such that $\mathbf{Z} - \mathbf{\Lambda}$ is a $\mathbf{F}$-local martingale (coordinate-wise). $\mathbf{\Lambda}$ is called the $\mathbf{F}$-**compensator** of $\mathbf{Z}$.*

*If there exists a non-negative, $\mathbf{F}$-predictable, locally integrable process $\boldsymbol{\lambda}$ satisfying, almost surely, $\Lambda_t = \int_0^t \lambda_s \mathrm{d}s$ for every $t \geqslant 0$, we call $\boldsymbol{\lambda}$ the $\mathbf{F}$-**stochastic intensity** of $\mathbf{Z}$.*

A well-known counting process is the *Poisson process* with rate $\alpha > 0$ (Protter, 2005, Chapter 3). Such a process $\mathbf{N}^\alpha = (N_t^\alpha)_{t\geqslant 0}$ has independent increments $N_t^\alpha - N_s^\alpha \sim \mathrm{Pois}(\alpha(t-s)), t > s \geqslant 0$, with $N_0^\alpha = 0$ a.s. A multivariate Poisson process $\mathbf{N}^\alpha$ with rate $\alpha = (\alpha_1, \dots, \alpha_d) \in (0, +\infty)^d$ has coordinate processes $\mathbf{N}^{\alpha_i}, 1 \leqslant i \leqslant d$ that are independent univariate Poisson processes with respective rate α_i. Its $\mathbf{F}^{\mathbf{N}^\alpha}$-stochastic intensity is the deterministic function $t \mapsto \alpha$.

Poisson processes have the Markov property: this result can be extended to any counting process $\mathbf{Z}$ whose $\mathbf{F}^{\mathbf{Z}}$-intensity has the form $t \mapsto \lambda_t = u_t(Z_{t-}) \in \mathbb{R}_+^d$, with u a locally integrable function; see Proposition 4. Simulation of $\mathbf{Z}$ is possible when its stochastic intensity has this form; see below.

Mixed Poisson processes. A natural extension of Poisson processes consists in replacing the constant rate by a random one. The resulting objects, called mixed Poisson processes, are extensively

covered by Grandell (1997), as well as Zocher (2005) in the multivariate case. The statements below are proved in Appendix B.2. In the rest of this section, the target distribution π is a probability distribution supported on $\mathbb{R}_+^d$, with finite first moment.

Definition 2. *Let X be a $\mathbb{R}_+^d$-valued random variable with distribution π. If, conditionally on X, $\mathbf{Z}$ is a Poisson process with rate X, we call $\mathbf{Z}$ the **mixed Poisson process (MPP)** with random rate X, and write $\mathbf{Z} \sim \mathrm{MPP}(\pi)$. In particular, conditionally on X, $Z_t \sim \mathrm{Pois}(tX)$ for any $t \geqslant 0$.*

A mixed Poisson process $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ with random rate X defines a *stochastic localization process* (Montanari, 2023): indeed, Theorem 1 shows that X is $\mathcal{F}_\infty^{\mathbf{Z}}$-measurable, while the Markov property of $\mathbf{Z}$ follows from Theorem 2.

Theorem 1. *Let $\mathbf{Z} \sim \mathrm{MPP}(\pi)$, with random rate X. Almost surely, $\lim_{t\to\infty} Z_t/t = X$.*

This result shows that if we were able to sample Z_t for a large t, Z_t/t would yield an approximate sample from π. Given our first definition of MPPs, sampling from $\mathbf{Z}$ requires sampling from π. Fortunately, similar to diffusion models (see Section 2.2), sample-free simulation of $\mathbf{Z}$ is possible if we have access to the corresponding *Poisson denoiser*. Indeed, Theorem 2, which generalizes the univariate case (Grandell, 1997, Proposition 4.1), shows that the $\mathbf{F}^{\mathbf{Z}}$-stochastic intensity of the mixed Poisson process coincides with its Poisson denoiser, defined as

$$(t, z) \in \mathbb{R}_+ \times \mathbb{N}^d \mapsto K_t^\pi(z) \coloneqq \mathbb{E}[X|Z_t = z] = \int_{\mathbb{R}_+^d} x\, \pi_t(\mathrm{d}x|z) \;, \tag{1}$$

where π_t is the conditional distribution of X given Z_t: for any $z \in \mathbb{N}^d$,

$$\pi_t(\mathrm{d}x|z) \propto \prod_{i=1}^d \mathrm{e}^{-tx_i} \frac{(tx_i)^{z_i}}{z_i!} \pi(\mathrm{d}x) \propto \prod_{i=1}^d \mathrm{e}^{-tx_i} x_i^{z_i} \pi(\mathrm{d}x) \;. \tag{2}$$

This denoiser depends solely on π and not on any particular $\mathbf{Z} \sim \mathrm{MPP}(\pi)$: in the univariate case, it can be expressed through the Laplace transform of π; see (Grandell, 1997, Chapter 4). More generally, we have the following result (Zocher, 2005, Chapter 3).

Proposition 2. *Given any $\mathbf{Z} \sim \mathrm{MPP}(\pi)$, the Poisson denoiser K^π defined by* (1) *can be expressed as follows. For every $t > 0$ and $z \in \mathbb{N}^d$,*

$$K_t^\pi(z) = \int_{\mathbb{R}_+^d} x \prod_{i=1}^d \mathrm{e}^{-tx_i} x_i^{z_i} \pi(\mathrm{d}x) \Bigg/ \int_{\mathbb{R}_+^d} \prod_{i=1}^d \mathrm{e}^{-tx_i} x_i^{z_i} \pi(\mathrm{d}x) \tag{3}$$

holds if $\mathbb{P}(Z_t = z) > 0$; otherwise, by convention, $K_t^\pi(z) = 0$. Furthermore, for any $z \in \mathbb{N}^d$, $t > 0 \mapsto K_t^\pi(z)$ is smooth and $t > 0 \mapsto \|K_t^\pi(z)\|_1$ is non-increasing.

Note that (3) allows us to extend the denoiser domain of definition to $\mathbb{R}_+ \times \mathbb{R}_+^d$.

Theorem 2. *The $\mathbf{F}^{\mathbf{Z}}$-compensator of a mixed Poisson process $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ is the process defined for every $t \geqslant 0$ by*

$$\Lambda_t = \int_0^t K_s^\pi(Z_s)\mathrm{d}s \;.$$

In other words, $\mathbf{Z}$ is a Markov counting process with $\mathbf{F}^{\mathbf{Z}}$-stochastic intensity $t \in \mathbb{R}_+ \mapsto K_t^\pi(Z_{t-})$.

The native Markov property of mixed Poisson processes is a notable difference from several Gaussian diffusion models. In these frameworks, one often considers a non-Markov process, and then constructs a Markov process with identical time-marginal distributions, the *Markovian projection*.

Learning the denoiser. Based on Theorem 2, we will see that $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ can be simulated if the denoiser K^π is known, or realistically approximated by an intensity function $(t, z) \mapsto k_t^\theta(z)$, yielding approximate samples from π. While K^π is generally intractable, it can be learnt since, by definition, we have $K_t^\pi(Z_t) = \mathbb{E}[X|Z_t]$ for any $t \geqslant 0$. This conditional expectation is the unique minimizer of any functional of the form $f \mapsto \mathbb{E}[\mathscr{D}_\phi(X, f(t, Z_t))]$, with $\phi : \mathbb{R}_+^d \to \mathbb{R}$ a strictly convex, differentiable function and $(x, y) \in \mathbb{R}_+^2 \mapsto \mathscr{D}_\phi(x, y) \coloneqq \phi(x) - \phi(y) - (x - y)^T \nabla\phi(y)$ is

the *Bregman divergence* associated with ϕ. A common choice is $x \mapsto \phi(x) = \|x\|_2^2$, which results in the squared Euclidean distance. To learn a Poisson denoiser, Bhattacharya et al. (2025); Chen & Zhou (2023) suggest using $x \mapsto \phi(x) = \mathrm{e}(x) = \sum_{i=1}^d x_i \log x_i$ instead, yielding the divergence

$$(x, y) \in (0, +\infty)^d \times (0, +\infty)^d \mapsto \mathscr{D}_{\mathrm{e}}(x, y) = \sum_{i=1}^d x_i \log(x_i/y_i) - x_i + y_i \ . \tag{4}$$

Given a choice of divergence $\mathscr{D}_\phi$ and a neural networks $(t, z) \mapsto k_t^\theta(z)$ parameterized by θ, we consider then the loss function $\theta \mapsto \mathscr{L}(\theta) = \int_0^T \mathbb{E}[\mathscr{D}_\phi(X, k_t^\theta(Z_t))]\mathrm{d}t$, with $T > 0$. The pseudo code associated with the minimization of this loss function is provided in Appendix D, Algorithm 3.

Simulation algorithms. We explain here how to simulate a counting process $\mathbf{Z}$ associated with an intensity of the form $t \mapsto u_t(Z_{t-})$. Proofs and additional details can be found in Appendix B.3.

Exact simulation of the process with such intensity is possible by adapting Ogata's *thinning* algorithm (Daley & Vere-Jones, 2008, Algorithms 7.5.IV, 7.5.V). It requires the intensity function's l^1 norm to be non-increasing in t, for any fixed z, which, by Proposition 2, holds notably for MPPs. This procedure and its validity are postponed to Appendix B.3, Proposition 5, Algorithm 2. This exact simulation method is a notable difference from Gaussian diffusion models, which necessarily induce a discretization error. However, the execution time of thinning is uncertain, as t increases according to exponentially distributed random variables. Furthermore, the execution time is expected to scale roughly linearly with the dimension, as each step increments only one coordinate.

Alternatively, we may use an algorithm with fixed execution time, at the cost of a discretization error. Figure 2 provides a visual representation of both simulation methods. For any $t > 0$, consider a time-grid $0 = t_0 < t_1 < \cdots < t_N = t$, with $\Delta_k \coloneqq t_{k+1} - t_k$. Define the Euler discretization $\tilde{\mathbf{Z}}$ of $\mathbf{Z}$ as the counting process with the following recursion: $\tilde{Z}_0 = 0$ and for every $k \geqslant 0, s \in [t_k, t_{k+1}]$, conditionally on $\tilde{Z}_{t_k}$, set

$$\tilde{Z}_s = \tilde{Z}_{t_k} + {}^k N_{s-t_k}^{u_{t_k}(\tilde{Z}_{t_k})} \ ,$$

where the ${}^k\mathbf{N}^{u_{t_k}(\tilde{Z}_{t_k})}$ is a standard Poisson process with constant intensity $u_{t_k}(\tilde{Z}_{t_k})$ and independent of the past. Said differently, $\tilde{\mathbf{Z}}$ has $\mathbf{F}^{\tilde{\mathbf{Z}}}$-stochastic intensity $s \mapsto \lambda_s \coloneqq \sum_{k=0}^{N-1} \mathbb{1}_{[t_k, t_{k+1})}(s) u_{t_k}(\tilde{Z}_{t_k})$. In particular, $\tilde{Z}_{t_{k+1}} - \tilde{Z}_{t_k} \sim \mathrm{Pois}(\Delta_k u_{t_k}(\tilde{Z}_{t_k}))$. We show in Proposition 6, Appendix B.2 that under mild regularity assumptions on u, $\mathbb{E}[\|\tilde{Z}_t - Z_t\|_1] = \mathcal{O}(\max_k \Delta_k)$ as $\max_k \Delta_k \to 0$. We also estimate the empirical discretization error on a simple example; see Figure 3.

Algorithm 1 Time-dicretization

Require: $(t, z) \mapsto u_t(z) \in \mathbb{R}_+^d$; time-grid $0 = t_0 < t_1 < ... < t_N$
1: Set $\tilde{Z}_0 = 0$
2: **for** $k = 0, \ldots, N-1$ **do**
3: Compute $u_{t_k}(\tilde{Z}_{t_k})$
4: Draw $n_k \sim \mathrm{Pois}(\Delta_k u_{t_k}(\tilde{Z}_{t_k}))$
5: Set $\tilde{Z}_{t_{k+1}} = \tilde{Z}_{t_k} + n_k$
6: **end for**
7: **return** $\tilde{Z}_{t_0}, \tilde{Z}_{t_1}, \cdots, \tilde{Z}_{t_N}$

Error bounds. In order to evaluate the quality of approximate π samples upon simulation of $\mathbf{Z}$, we first derive an *early stopping error*, due to the restriction of $\mathbf{Z}$ to a finite time-window.

Lemma 1. *Let $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ with random rate X. Assume $\mathbb{E}[\|X\|_2^2] < \infty$; then for every $t > 0$,*

$$\mathscr{W}_2(\mathrm{Law}(Z_t/t), \pi)^2 \leqslant \mathbb{E}[\|Z_t/t - X\|_2^2] = \mathbb{E}[\|X\|_1]/t \ ,$$

where $\mathscr{W}_2$ is the 2-Wasserstein distance defined in Appendix A, (8).

Compared to the analogous bound for Gaussian diffusion (Section 2.2), there is an additional dependence on $\mathbb{E}[\|X\|_1]$ due to the equidispersion of the Poisson distribution. This result is reminiscent of (Saremi et al., 2024, Proposition 2), where X is estimated from a discrete set of measurements.

Remark 1. *Instead of approximating X by Z_t/t for large t, using the denoiser $\mathbb{E}[X|Z_t]$ instead is preferable; see also Figure 4. Indeed, a short computation shows that*

$$\mathbb{E}\left[\|X - \mathbb{E}[X|Z_t]\|_2^2\right] = \mathbb{E}\left[\|X - Z_t/t\|_2^2\right] - \mathbb{E}\left[\|\mathbb{E}[X|Z_t] - Z_t/t\|_2^2\right] \ .$$

Thus $\mathbb{E}[X|Z_t] \to X$ in L^2, with an early stopping error no larger than the one obtained using Z_t/t.

In our next result, we show that the distributions of two counting processes can be quantitatively controlled in terms of the integrated discrepancy between their respective intensity functions, provided one is Lipschitz. As a direct consequence, we obtain a bound on the distance between the law of an MPP targeting π and that of an approximate MPP driven by an intensity $k^\theta \approx K^\pi$.

Lemma 2. *Let $\mathbf{Z}, \mathbf{Y}$ be counting processes with respective intensities $t \mapsto u_t(Z_{t-}), t \mapsto v_t(Y_{t-})$, where u, v are measurable, locally integrable functions. Assume that for $t > 0$, the integrated discrepancy $\delta(t) \coloneqq \int_0^t \mathbb{E}[\|u_s(Y_s) - v_s(Y_s)\|_1]\mathrm{d}s$ is finite, and finally that $t \mapsto \mathrm{Lip}(u_t) = \sup_{y \neq z} \|u_t(z) - u_t(y)\|_1 / \|z - y\|_1$ is locally integrable. Then, for $t > 0$,*

$$\mathscr{W}_{1,1}\left(\mathrm{Law}\left(Z_t\right), \mathrm{Law}\left(Y_t\right)\right) \leqslant \delta(t) \exp\left(\int_0^t \mathrm{Lip}(u_s)\mathrm{d}s\right) ,$$

where $\mathscr{W}_{1,1}$ denotes the 1-Wasserstein distance associated to the l^1 distance: see Appendix A, (9).

The bound derived in Lemma 2 is relatively sharp: in Example 3, we consider a tractable case where π is Gamma distributed on $\mathbb{R}_+$, and show that the upper bound differs from the actual Wasserstein distance by only a logarithmic factor in t. Finally, combining the results above, we can establish, under appropriate conditions, theoretical guarantees on the sampling performance of MPDM; see Appendix B.3, Corollary 1.

2.2 Related work and comparison with diffusion models

Comparison with classical Gaussian diffusion models. We choose here the same diffusion noising process as, e.g., Grenioux et al. (2024). While many diffusion models are parametrized through their score, Tweedie's formula shows that one can equivalently work with the denoiser; with this construction, the denoiser appears naturally, which is reminiscent of the MPDM construction of the previous section. Consider a target distribution π on $\mathbb{R}^d$, $X \sim \pi$, and the process $\mathbf{X}^{\mathrm{n}} = (X_t^{\mathrm{n}})_{t \geqslant 0}$ defined as $X_t^{\mathrm{n}} = tX + W_t$, where $\mathbf{W}$ is a d-dimensional Brownian motion independent of X. Similar to mixed Poisson processes, we have $X_t^{\mathrm{n}}/t \to X$ when $t \to \infty$ almost surely. Given $t > 0$, we denote the conditional distribution of X given X_t^{n} as $\pi_t(\mathrm{d}x|x_t) \propto \mathrm{e}^{-\|x_t - tx\|^2/(2t)}\pi(\mathrm{d}x)$, and define the *Gaussian denoiser* as the function $(t, x_t) \in \mathbb{R}_+ \times \mathbb{R}^d \mapsto D_t^\pi(x_t) = \int x \times \pi_t(\mathrm{d}x|x_t)$.

A priori, it is not possible to simulate this process without first sampling from π. However, assuming π has a finite first moment, the solution of the stochastic differential equation

$$\mathrm{d}X_t^{\mathrm{d}} = D_t^\pi(X_t^{\mathrm{d}})\mathrm{d}t + \mathrm{d}B_t \ , \tag{5}$$

where $\mathbf{B}$ is another Brownian motion, is a Markov process with the same time-marginal distributions as $\mathbf{X}^{\mathrm{n}}$. The Gaussian denoiser can be learned by minimization of a functional, like the Poisson denoiser. Samples from the joint distribution (X, X_t^{n}) are obtained by 1) drawing $x \sim \pi$ from a dataset and 2) computing the noisy observation $x_t \sim X_t^{\mathrm{n}}$ as $tx + \sqrt{t}G$, with $G \sim \mathrm{N}(0, \mathrm{I}_d)$. Once the denoiser is learned, we obtain approximate samples from π by simulating (5) on a discretized time grid. However, diffusion models are not limited to generative modeling tasks: a pre-trained model may also be used to solve inverse problems, which is the object of Section 3.

Related works. Poissonian generative models were previously considered by Chen & Zhou (2023); Bhattacharya et al. (2025), but their constructions differ substantially from ours.

Chen & Zhou (2023) introduced JUMP, which encodes X into a Poisson-distributed latent variable $Z_{t_0} \sim \mathrm{Pois}(\lambda X)$, constructs a discrete-time Markov chain $0 \approx Z_{t_N}, Z_{t_{N-1}}, \ldots, Z_{t_1}, Z_{t_0}$ through a binomial thinning argument, and derives the corresponding training objective from a variational lower bound. Their resulting training and sampling procedures (*i.e.*, inference algorithm) are closely related to our Algorithms 1 and 3 when the Bregman divergence (4) is used to train the denoiser; see Appendix D.1 for further details. The main algorithmic differences between JUMP and MPDM are analogous to those between DDPMs (Ho et al., 2020) and SDE-based diffusion models (Song et al., 2021). Beyond these algorithmic similarities, the continuous-time mathematical framework developed here also enables us to derive theoretical guarantees for MPDMs.

Bhattacharya et al. (2025) adopt an information-theoretic perspective, in the spirit of the diffusion framework developed by Kong et al. (2023). Conditionally on X, they consider, for any *fixed* $t \geqslant 0$, an observation channel of the form $Z_t \sim \mathrm{Pois}(tX)$. They then propose learning the Poisson denoiser

$\mathbb{E}[X|Z_t]$ through a reconstruction loss related to the ELBO derived in (Chen & Zhou, 2023). They also introduce a sampling procedure that differs substantially from both JUMP and our method. Through a controlled synthetic experiment, we show in Appendix B.3, Remark 2 that this inference procedure appears less effective than the one considered in Algorithm 1.

3 Poisson Bayesian inverse problems

We next apply the MPDM framework to inverse problems. We review recent diffusion-based methods that leverage a pre-trained diffusion model in Appendix C. These approaches have been successfully applied to diverse inverse problems, but mostly ones driven by Gaussian noise. In this section, we consider Poisson inverse problems of the form $Y \sim \mathrm{Pois}(\mathcal{A}(X))$, with forward operator $\mathcal{A} : \mathbb{R}_+^d \to \mathbb{R}_+^{d_y}$. The objective is to sample from $\pi^y \propto g(y|x)\pi(\mathrm{d}x)$, the posterior distribution of X given $Y = y$, with prior distribution π supported on $\mathbb{R}_+^d$ and likelihood function g defined in (26).

We show how a pretrained denoiser $k^\theta \approx K^\pi$ associated with $\mathrm{MPP}(\pi)$ can be used to approximately sample from $\mathrm{MPP}(\pi^y)$, hence from the posterior distribution π^y, in close analogy with diffusion-based posterior sampling methods. In particular, we adapt DPS (Chung et al., 2023), PnP-DM (Wu et al., 2024) and DAPS (Zhang et al., 2025) to the MPDM framework.

Base case: Poisson denoising. We first consider the Poisson inverse problem $Y \sim \mathrm{Pois}(aX)$, $a > 0$, *i.e.*, $\mathcal{A} = a\mathrm{I}_d$. Given an observation $Y = y$, Theorem 3 shows that the *conditional* denoiser K^{π^y} of $\mathbf{Z}^y \sim \mathrm{MPP}(\pi^y)$ can be obtained through a shift of K^π inputs. Thus, simulating $\mathbf{Z}^y$ requires neither additional approximation nor retraining. We refer to the associated posterior sampling method as *Poisson Denoising Posterior Sampling* (PDPS); see Algorithm 4. This result is the Poisson analogue of the "Gaussian likelihood, Gaussian diffusion" case; see Appendix C, (25).

Theorem 3. *Let π^y denote the posterior distribution of X given $Y = y$, where $Y \sim \mathrm{Pois}(aX)$ and $X \sim \pi$. The denoiser K^{π^y} associated with $\mathrm{MPP}(\pi^y)$ is given by $(t, z) \mapsto K_t^{\pi^y}(z) = K_{t+a}^\pi(z + y)$.*

We next turn to general forward operators and show how several diffusion-based posterior sampling algorithms can be adapted to the MPDM framework. In this section, we state only the main results underlying these adaptations; implementation details and proofs are deferred to Appendix D. Although our main motivation is Poisson inverse problems, the remainder of this section does not require any specific assumption on the likelihood g and therefore extends beyond Poisson inverse problems; we only require that the prior π be supported on $\mathbb{R}_+^d$.

A DPS-like algorithm. Consider a general forward operator $\mathcal{A}$ and the associated likelihood g. Recall that $\pi_t(\mathrm{d}x|z_t)$ denotes the conditional distribution of X given $Z_t = z_t$, for $\mathbf{Z} \sim \mathrm{MPP}(\pi)$, as defined in (2). Define $g_t(y|z_t) \coloneqq \int_{\mathbb{R}_+^d} g(y|x)\pi_t(\mathrm{d}x|z_t)$. The following result gives an exact expression for the posterior denoiser in terms of a discrete likelihood-ratio correction.

Theorem 4. *Consider the posterior distribution π^y of a general Poisson inverse problem with likelihood function g defined in* (26)*. For any $z_t \in \mathbb{N}^d$ and $t > 0$, writing $e_i = (\delta_{i,j})_{1 \leqslant j \leqslant d}$, we have*

$$[K_t^{\pi^y}(z_t)]_i = [K_t^\pi(z_t)]_i \frac{g_t(y|z_t + e_i)}{g_t(y|z_t)} = [K_t^\pi(z_t)]_i \left(1 + \frac{g_t(y|z_t + e_i) - g_t(y|z_t)}{g_t(y|z_t)}\right) , 1 \leqslant i \leqslant d \, .$$

This expression is reminiscent of its diffusion counterpart (see Appendix C, (22)), with a *discrete derivative* replacing the usual continuous gradient. Leveraging the usual DPS Dirac approximation $\pi_t(\mathrm{d}x|z_t) \approx \delta_{K_t^\pi(z_t)}(\mathrm{d}x)$, then approximating the discrete derivative by a continuous one, we get

$$\begin{aligned} [K_t^{\pi^y}(z_t)]_i &\approx [K_t^\pi(z_t)]_i \left(1 + \frac{g(y|K_t^\pi(z_t + e_i)) - g(y|K_t^\pi(z_t))}{g(y|K_t^\pi(z_t))}\right) \\ &\approx [K_t^\pi(z_t)]_i \left(1 + \partial_i \log g(y|K_t^\pi(z_t))\right) , \end{aligned} \tag{6}$$

where the domain of $K_t^\pi(\cdot)$ is extended to $\mathbb{R}_+^d$ using the expression of Proposition 2, and also implicitly assuming its differentiability. Note that non-negativity of the denoiser is not necessarily guaranteed with this approximation: in practice, we project it to the positive orthant. Alternatively, one may retain the discrete derivative and approximate (6) by evaluating $K_t^\pi(z_t + \mathbf{1}) - K_t^\pi(z_t)$,

thereby implicitly assuming that $[K_t^\pi(z_t)]_i$ depends only on z_t^i. Our empirical results suggest that the continuous-derivative approximation performs better at large t, whereas the "independent coordinates" approximation can be slightly more accurate at small t; see Appendix D.3 for additional details. Combining either approximation of K^{π^y} with the simulation scheme of Algorithm 1 yields our MPDM analogue of DPS. Pseudocode is provided in Algorithm 5.

Example 1. *From Appendix B.4, when $\pi = \gamma = \mathbf{\Gamma}(\alpha, \beta)$, $\alpha, \beta \in (0, +\infty)^d$, we have $[K_t^\gamma(z_t)]_i = (z_t^i + \alpha_i)/(\beta_i + t), 1 \leqslant i \leqslant d$. In this case, both derivative approximations are exact.*

The ΠGDM algorithm (Song et al., 2023) relies on an approximation different from that used in DPS. The denoising distribution is approximated by a Gaussian distribution. For Gaussian inverse problems with linear $\mathcal{A}$, this approximation makes the conditional score, or equivalently the conditional denoiser, tractable. However, ΠGDM does not directly apply to Poisson inverse problems.

Split Gibbs sampling methods. We adapt PnP-DM by introducing the augmented distributions

$$\pi_k^y(\mathrm{d}x, \mathrm{d}\tilde{x}) \propto g(y|\tilde{x}) \exp\left(-\mathscr{D}_\mathrm{e}(\tilde{x}, x)/\rho_k^2\right) \pi(\mathrm{d}x) \,, \tag{7}$$

which mirror the structure of (23), where $\mathscr{D}_\mathrm{e}$ denotes the Bregman divergence defined in (4). The prior step, namely sampling from $\pi_k^y(\mathrm{d}x|\tilde{x})$, can be performed using $\mathrm{MPP}(\pi)$. More precisely, $\pi_k^y(\mathrm{d}x|\tilde{x}) \propto \pi_{t_k}(\mathrm{d}x\,|t_k\tilde{x})$ with $t_k \coloneqq 1/\rho_k^2$; see Proposition 3. The likelihood step, that is, sampling from $\pi_k^y(\mathrm{d}\tilde{x}|x)$, requires an MCMC procedure. To handle the positivity constraint, we use a Riemannian Langevin method; see Algorithm 7, Appendix D.4, and (Girolami & Calderhead, 2011).

Proposition 3. *For every $t > 0$ and $z_t \in \mathbb{N}^d$, $\pi_t(\mathrm{d}x|z_t) \propto \exp\left(-t\mathscr{D}_\mathrm{e}(z_t/t, x)\right) \pi(\mathrm{d}x)$.*

To adapt DAPS to our setting, we approximate the denoising distribution as $\pi_t(\mathrm{d}x|z_t) \approx \mathbf{\Gamma}(r_t^{-2}\hat{x}(z_t), r_t^{-2}1)$, where typically $\hat{x}(z_t) = K_t^\pi(z_t)$. When π is Gamma distributed, the denoising distribution is itself Gamma distributed and is exactly of this form, providing a natural justification for the approximation; see Appendix D.5. As in the original DAPS method, $\hat{x}(z_t)$ may alternatively be estimated by simulating the MPP forward from z_t. The remaining steps of the algorithm can then be adapted directly to the MPDM setting; see Appendix D.5, Algorithm 8.

4 Poisson inverse problems: numerical experiments

Despite our focus on inverse problems, we emphasise that MPDMs are generative models, and that prior works on Poissonian generative modelling suggest potential advantages over classical diffusion models for certain data distributions; see (Chen & Zhou, 2023; Bhattacharya et al., 2025). For the experiments below, we trained MPDMs on the *BrainWeb* dataset (Aubert-Broche et al., 2006; Schramm, 2021), and mock galaxy images from the *EAGLE* simulations (McAlpine et al., 2016); see Appendix E.5 for a gallery of MPDM-generated images, and Appendix E.1 for datasets details.

Benchmark. Several diffusion-based posterior sampling algorithms are formulated for additive, homogeneous Gaussian noise, and do not directly apply to Poisson inverse problems, even with Gaussian approximation $\mathrm{Pois}(\mathcal{A}(X)) \approx \mathrm{N}(\mathcal{A}(X), \mathrm{diag}(\mathcal{A}(X)))$. DPS can accommodate both the true Poisson likelihood and its Gaussian approximation; the latter is the canonical version described by Chung et al. (2023). PnP-DM and DAPS likelihood steps can theoretically handle a Poisson likelihood. Our benchmark can be separated into three blocks. 1) Our MPDM-based algorithms adapting DPS (Algorithm 5), PnP-DM (Algorithm 6) and DAPS (Algorithm 8) for general inverse problems; or PDPS (Algorithm 4) for the Poisson denoising problems. 2) Diffusion-based models with the exact Poisson likelihood: DPS, PnP-DM and DAPS; or 3) with Gaussian-approximated likelihood. Each method uses the same number of NFEs (*i.e.*, pre-trained denoiser calls). Computational costs nevertheless differ, notably because PnP-DM and DAPS require MCMC steps; see Appendix E for details. We train MPDMs and Gaussian diffusion-based models from scratch for each dataset, with identical architectures and training budgets; see Appendix E.3.

The performance on realistic Poisson inverse problems is evaluated through several metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), as well as Learned Perceptual Image Patch Similarity (LPIPS). These metrics evaluate the quality of a single reconstructed image, and Table 2 shows averages over the validation datasets. We provide a complementary evaluation of posterior sampling quality in Appendix E.4.

Methods	EAGLE denoising (LC)			EAGLE denoising (HC)			BrainWeb denoising (LC)			BrainWeb denoising (HC)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
PDPS (MPDM)	32.34	0.7916	0.1543	34.94	0.8435	0.0863	16.90	0.7812	0.1124	21.31	0.9019	0.0585
DPS (MPDM)	31.55	0.74729	0.1783	33.16	0.7748	0.1369	16.91	0.7574	0.1429	20.49	0.8477	0.1207
PnP-DM (MPDM)	26.25	0.5933	0.4149	34.80	0.8401	0.0991	16.65	0.6278	0.2258	21.72	0.8171	0.1408
DPS-p (diffusion)	32.46	0.8060	0.2213	34.69	0.8409	0.1612	17.22	0.7915	0.0890	20.62	0.8814	0.0591
DAPS-p (diffusion)	29.92	0.7268	0.2547	34.60	0.8388	0.1546	17.12	0.7794	0.1141	21.41	0.8796	0.0668
Methods	**EAGLE blur (LC)**			**EAGLE blur (HC)**			**BrainWeb PET (LC)**			**BrainWeb PET (HC)**		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DPS (MPDM)	31.13	0.7638	0.2049	31.64	0.7764	0.1858	15.56	0.7084	0.1395	16.84	0.7503	0.1318
PnP-DM (MPDM)	29.83	0.7341	0.3039	33.22	0.8000	0.2511	16.41	0.7329	0.1051	18.11	0.7940	0.0828
DPS-p (diffusion)	31.72	0.7935	0.2432	32.56	0.8017	0.2129	15.82	0.7401	0.0968	16.66	0.7758	0.0947
DAPS-g (diffusion)	31.99	0.7881	0.2652	33.14	0.8041	0.2365	16.13	0.7478	0.1111	17.60	0.7930	0.1115

Table 2: Quality of reconstructed images for different posterior sampling algorithms and Poisson inverse problems. Colors show best and second best, with ties due to overlapping confidence intervals. We only show best-performing methods: complete results are deferred to Appendix E.3. DPS-p and DAPS-p use the Poisson likelihood, and DAPS-g the Gaussian-approximated one.

Toy example. To demonstrate the capacity of PDPS to accurately sample from the Bayesian posterior π^y associated with the Poisson denoising problem ($\mathcal{A} = a\mathrm{I}_d$), we consider a fully tractable example, where π is a mixture of Gamma distributions; see Appendix B.4 for details. We train a Poisson and a Gaussian generative model; although the Poisson model is tractable in this setting, we learn it to ensure fairness. We draw 10 observations y in both a low-count (LC) and a high-count (HC) regime ($a = 0.6$ and $a = 3$, respectively), consider the associated π^y, and evaluate the sampling performance of different algorithms. We report on Tables 1 and 4 the 2-Wasserstein distance between exact π^y samples and approximate ones, averaged over the observations y. PDPS significantly outperforms its competitors: indeed, except for the "intrinsic" errors described in Section 2.1, this method introduces no additional sampling bias.

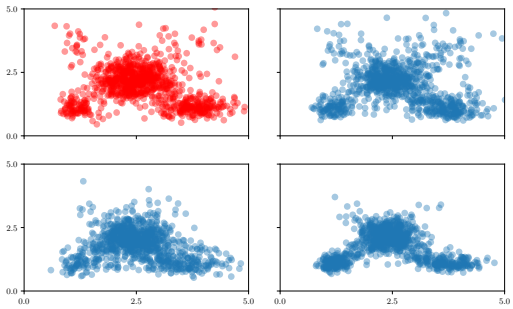

Figure 1: True and approximate π^y samples, obtained with PDPS (top-right), diffusion DPS (bottom-left) and PnP-DM (bottom-right).

Methods	Toy (LC)	Toy (HC)
PDPS (MPDM)	0.0945 ± 0.0053	0.0732 ± 0.0023
DPS-p (diffusion)	0.2134 ± 0.0046	0.2999 ± 0.0038

Table 1: Average 2-Wasserstein distance between π^y exact samples and approximate ones.

Poisson inverse problems. We consider two classical Poisson inverse problems. PET (Bailey et al., 2005), where $\mathcal{A}$ is a Radon operator, and astronomical imaging, where $\mathcal{A}$ is a Gaussian blur operator (Donath et al., 2024), as well as the denoising problem ($\mathcal{A} = a\mathrm{I}_d$). Each problem is studied in a both low-count regime and a high-count regime. For Poisson denoising problems, despite its low computational budget, PDPS consistently ranks first or second. MPDM-based DPS yields the best LPIPS for the blur problems, and MPDM-based PnP-DM the best overall performance for PET. Diffusion-based DPS and DAPS remain competitive.

5 Conclusion

We construct and study "Poissonian" diffusion models, named MPDMs, using the multivariate counting processes framework. We motivate the use of MPDM beyond generative modeling by adapting several diffusion-based posterior sampling algorithms; in particular, we demonstrate mathematically and empirically that MPDMs are well adapted to Poisson inverse problems, which remain relatively under-explored in this literature.

AI USE STATEMENT

Generative AI tools were used to assist with minor English-language editing and polishing of parts of the main text. They were also used to assist with coding during the implementation and experimentation, with the resulting code systematically verified by the authors. All mathematical content, including the derivation of the theoretical results, was developed by the authors without the use of generative AI tools.

REPRODUCIBILITY STATEMENT

Appendices B and D contain detailed proofs of the mathematical statements as well as the pseudo-code associated with every algorithm mentioned in the main text. Implementation details are reported in Appendix E. Both datasets used in Section 4 are public; see also Appendix E.1.

REFERENCES

Berengère Aubert-Broche, Mark Griffin, G. Bruce Pike, Alan C. Evans, and D. Louis Collins. Twenty new digital brain phantoms for creation of validation image data bases. *IEEE transactions on medical imaging*, 25(11):1410–1416, November 2006. ISSN 0278-0062. doi: 10.1109/TMI.2006.883453.

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*, 2021.

Dale L Bailey, David W Townsend, Peter E Valk, and Michael N Maisey (eds.). *Positron Emission Tomography: Basic Sciences*. Springer, London, 2005. ISBN 978-1-85233-798-8 978-1-84628-007-8. doi: 10.1007/b136169. URL https://link.springer.com/10.1007/b136169.

Sagnik Bhattacharya, Abhiram Gorle, Ahsan Bilal, Connor Ding, Amit Kumar Singh Yadav, and Tsachy Weissman. Itdpdm: Information-theoretic discrete poisson diffusion model. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 10860–10915. Curran Associates, Inc., 2025. doi: 10.52202/085713-0364. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1021014b8c5a9008fc245736a1b73d7e-Paper-Conference.pdf.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, 2022.

Tianqi Chen and Mingyuan Zhou. Learning to Jump: Thinning and Thickening Latent Counts for Generative Modeling. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 5367–5382. PMLR, July 2023. URL https://proceedings.mlr.press/v202/chen23ap.html. ISSN: 2640-3498.

Julien Chevallier. *Modelling large neural networks via Hawkes processes*. Theses, COMUE Université Côte d'Azur (2015 - 2019), September 2016. URL https://theses.hal.science/tel-01415622. Issue: 2016AZUR4051.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Probability and Its Applications. Springer, New York, NY, 2008. ISBN 978-0-387-21337-8 978-0-387-49835-5. doi: 10.1007/978-0-387-49835-5. URL http://link.springer.com/10.1007/978-0-387-49835-5.

Axel Donath, Aneta Siemiginowska, Vinay L. Kashyap, David A. van Dyk, and Douglas Burke. Joint Deconvolution of Astronomical Images in the Presence of Poisson Noise. *The Astronomical Journal*, 168(4):182, September 2024. ISSN 1538-3881. doi: 10.3847/1538-3881/ad6b98. URL https://doi.org/10.3847/1538-3881/ad6b98. Publisher: The American Astronomical Society.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2): 123–214, 2011. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2010.00765.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2010.00765.x.

J. Grandell. *Mixed Poisson Processes*. Chapman and Hall/CRC, Boca Raton, 1997. ISBN 978-1-003-05995-0. doi: 10.1201/9781003059950.

Louis Grenioux, Maxence Noble, Marylou Gabrié, and Alain Oliviero Durmus. Stochastic localization via iterative posterior sampling. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=2Gr5wZR6uc.

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

Yazid Janati, Badr MOUFAD, Mehdi Abou El Qassime, Alain Oliviero Durmus, Eric Moulines, and Jimmy Olsson. A mixture-based framework for guiding diffusion models. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=HIOA3bScFB.

Yazid Janati, Eric Moulines, Jimmy Olsson, and Alain Oliviero-Durmus. Bridging diffusion posterior sampling and Monte Carlo methods: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 383(2299):20240331, June 2025b. ISSN 1364-503X. doi: 10.1098/rsta.2024.0331. URL https://doi.org/10.1098/rsta.2024.0331.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration 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=kxXvopt9pWK.

Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UvmDCdSPDOW.

Andreas E. Kyprianou. *Fluctuations of Lévy Processes with Applications: Introductory Lectures*. Universitext. Springer, Berlin, Heidelberg, 2014. ISBN 978-3-642-37631-3 978-3-642-37632-0. doi: 10.1007/978-3-642-37632-0. URL https://link.springer.com/10.1007/978-3-642-37632-0.

Patrick J. Laub, Thomas Taimre, and Philip K. Pollett. Hawkes Processes, July 2015. URL http://arxiv.org/abs/1507.02822. arXiv preprint.

S. McAlpine, J. C. Helly, M. Schaller, J. W. Trayford, Y. Qu, M. Furlong, R. G. Bower, R. A. Crain, J. Schaye, T. Theuns, C. Dalla Vecchia, C. S. Frenk, I. G. McCarthy, A. Jenkins, Y. Rosas-Guevara, S. D. M. White, M. Baes, P. Camps, and G. Lemson. The eagle simulations of galaxy formation: Public release of halo and galaxy catalogues. *Astronomy and Computing*, 15:72–89, 2016. ISSN 2213-1337. doi: https://doi.org/10.1016/j.ascom.2016.02.004. URL https://www.sciencedirect.com/science/article/pii/S2213133716300130.

Andrea Montanari. Sampling, Diffusions, and Stochastic Localization, May 2023. URL https://arxiv.org/abs/2305.10690v2. arXiv preprint.

Philip E. Protter. *Stochastic Integration and Differential Equations*, volume 21 of *Stochastic Modelling and Applied Probability*. Springer, Berlin, Heidelberg, 2005. ISBN 978-3-642-05560-7 978-3-662-10061-5. doi: 10.1007/978-3-662-10061-5. URL http://link.springer.com/10.1007/978-3-662-10061-5.

O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pp. 234–241. Springer, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a. (available on arXiv:1505.04597 [cs.CV]).

Saeed Saremi, Ji Won Park, and Francis Bach. Chain of Log-Concave Markov Chains. *International Conference on Learning Representations*, 2024:17499–17522, May 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4b7439a4ab0b8e4bcb4e2412c6a10a58-Abstract-Conference.html.

Georg Schramm. Simulated brainweb PET/MR data sets for denoising and deblurring. June 2021. URL https://zenodo.org/records/4897350.

Georg Schramm and Kris Thielemans. PARALLELPROJ—an open-source framework for fast calculation of projections in tomography. *Frontiers in Nuclear Medicine*, Volume 3 - 2023, 2024. ISSN 2673-8880. doi: 10.3389/fnume.2023.1324562. URL https://www.frontiersin.org/journals/nuclear-medicine/articles/10.3389/fnume.2023.1324562.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine Bouman. Principled probabilistic imaging using diffusion models as plug-and-play priors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Xq9HQf7VNV.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20895–20905, 2025.

Mathias Zocher. *Multivariate Mixed Poisson Processes*. PhD thesis, Technische Universität Dresden, 2005.

A Further notation

Throughout this paper, $\mathbb{N} = \{0, 1, \dots\}$ denotes the set of natural numbers, $\mathbb{R}$ the set of real numbers and $\mathbb{R}_+ \coloneqq [0, +\infty($ the set of non-negative real numbers. Given $d \geqslant 1$, we write a d-dimensional vector as $x = (x_1, \dots, x_d) \in \mathbb{R}^d$. $\|x\|_2$ denotes the usual Euclidean norm, and $\|x\|_1 = \sum_{i=1}^d |x_i|$ the l^1 norm.

Given two probability distributions μ, ν on $\mathbb{R}^d$, the 2-Wasserstein distance is defined as

$$\mathscr{W}_2(\mu, \nu) = \inf_{\rho \in \mathscr{C}(\mu,\nu)} \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2^2 \, \rho(\mathrm{d}x, \mathrm{d}y) \right)^{1/2}, \tag{8}$$

where $\mathscr{C}(\mu, \nu)$ denotes the set of probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with first (respectively second) marginal equals to μ (respectively ν). In particular, if $X \sim \mu, Y \sim \nu$, then $\mathscr{W}_2(\mu, \nu)^2 \leqslant \mathbb{E}\left[\|X - Y\|_2^2\right]$. In this work, we also use the 1,1-Wasserstein distance, which we define as

$$\mathscr{W}_{1,1}(\mu, \nu) = \inf_{\rho \in \mathscr{C}(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_1 \, \rho(\mathrm{d}x, \mathrm{d}y) \, . \tag{9}$$

The exponential distribution with rate $\lambda > 0$ is denoted by $\mathrm{Exp}(\lambda)$, and $\mathrm{Unif}_{[a,b]}$ is the uniform distribution on $[a, b] \subseteq \mathbb{R}$. The d-dimensional normal distribution with mean $\mu \in \mathbb{R}^d$ and isotropic variance $\sigma^2 > 0$ is denoted by $\mathrm{N}(\mu, \sigma^2 \mathrm{I}_d)$, where I_d is the identity matrix in dimension d. If $\alpha \in \mathbb{R}_+^d$, the multivariate Poisson distribution with mean α is denoted by $\mathrm{Pois}(\alpha)$: a $\mathbb{N}^d$-valued random variable $Z \sim \mathrm{Pois}(\alpha)$ has independent coordinates $Z_i \sim \mathrm{Pois}(\alpha_i)$, with $\mathrm{Pois}(\alpha_i)$ the usual univariate Poisson distribution with mean $\alpha_i \in \mathbb{R}_+$. Similarly, given $\alpha, \beta \in (0, +\infty)^d$, we write the multivariate Gamma distribution (with independent coordinates) as $\mathbf{\Gamma}(\alpha, \beta)$. Finally, if $x \in \mathbb{R}_+^d \setminus \{0\}$, the categorical distribution $\mathrm{Cat}(x)$ is the probability distribution on $\{1, \dots, d\}$ defined by $\mathrm{Cat}(x)(\{i\}) = x_i / \|x\|_1$.

B Mixed Poisson processes: additional content

This section is an extended version of Section 2, with additional statements, detailed proofs, and illustrations.

B.1 Background on counting processes

Point processes. In Definition 1, we define counting processes as time-indexed stochastic processes with unit-jump trajectories, and adopt a martingale approach to characterize them through their compensator (or stochastic intensity). While this presentation suffices to develop our generative framework and is furthermore analogous to SDE-based diffusion models, counting processes are usually part of the *point processes* literature. We provide a brief overview of this topic and refer to (Daley & Vere-Jones, 2008) for a detailed introduction.

A *point process* (Daley & Vere-Jones, 2008, Chapter 9) ξ is an integer-valued random measure on an adequate measurable metric space X equipped with its Borel σ-algebra $\mathcal{B}(\mathsf{X})$. To be precise, for any $\omega \in \Omega$, $\mathsf{A} \in \mathcal{B}(\mathsf{X}) \mapsto \xi(\omega)(\mathsf{A}) \in \mathbb{N}$ is an integer-valued, locally finite measure on X; for a fixed $\mathsf{A} \in \mathcal{B}(\mathsf{X})$, $\omega \in \Omega \mapsto \xi(\omega)(\mathsf{A}) \in \mathbb{N}$ is an integer-valued random variable.

A *simple* point process satisfies, for almost all $\omega \in \Omega$, $\xi(\omega)(\{t\}) \in \{0, 1\}$ for any $t \in \mathsf{X}$, and can be written as a purely atomic measure: almost surely, there exists a countable set $\{t_i\}_i$, such that $\{t_i\} \cap \mathsf{A}$ is finite for any compact $\mathsf{A} \in \mathcal{B}(\mathsf{X})$, and

$$\xi = \sum_i \delta_{t_i} \ .$$

Univariate counting processes can be viewed as simple point processes on $\mathbb{R}_+$: given a counting process $\mathbf{Z}$ with jump times $(T_i)_{i \geqslant 1}$, $\xi = \sum_{i \geqslant 1} \delta_{T_i}$ is a simple point process on $\mathbb{R}_+$. Reciprocally, given a simple point process on $\mathbb{R}_+$ ξ, $t \mapsto \xi([0, t])$ is a counting process.

A simple *marked point process* is obtained by replacing the ground space X with, typically, $\mathsf{X} \times \{1, \dots, d\}$:

$$\xi = \sum_i \delta_{(t_i, k_i)} \ ,$$

with $k_i \in \{1, \dots, d\}$. A multivariate counting process $\mathbf{Z}$ can be viewed as a marked point process on $\mathbb{R}_+ \times \{1, \dots, d\}$, writing, for $1 \leqslant k \leqslant d$, $t \geqslant 0$, $Z_t^k = \xi([0, t] \times \{k\})$.

A useful special case is the following: Π is said to be a *Poisson point process with intensity measure* μ, with μ a measure on X, if it satisfies $\mathrm{Law}(\Pi(\mathsf{A})) = \mathrm{Pois}(\mu(\mathsf{A}))$ for every measurable $\mathsf{A} \in \mathcal{B}(\mathsf{X})$; and if furthermore $\Pi(\mathsf{A}_1), \dots, \Pi(\mathsf{A}_n)$ are independents when $\mathsf{A}_1, \dots, \mathsf{A}_n$ are disjoint measurable sets.

We may use a Poisson point process on $\mathbb{R}_+ \times \mathbb{R}_+$ to build a univariate counting process $\mathbf{Z}$ from its $\mathbf{F}$-stochastic intensity $\boldsymbol{\lambda}$. Consider a $\mathbf{F}$-Poisson point process Π with intensity measure $\mathrm{d}t\mathrm{d}x$ on $\mathbb{R}_+ \times \mathbb{R}_+$, *i.e.*, for every $\mathsf{A} \in \mathcal{B}(\mathbb{R}_+)$, $\Pi([0, t] \times \mathsf{A})$ is a $\mathbf{F}$-adapted Poisson process. As a side note, the $\mathbf{F}$-Poisson point process has the strong Markov property: conditionally on $\mathcal{F}_T$, with T a $\mathbf{F}$-stopping time, the restriction of Π to $[T, +\infty) \times \mathbb{R}_+$ is still a Poisson point process, independent of $\mathcal{F}_T$.

Following (Chevallier, 2016, Theorem 2.8.11) or (Daley & Vere-Jones, 2008, Proposition 7.5.I), if $\mathbf{F}^{\mathbf{Z}} \subseteq \mathbf{F}$, we have for every $t \geqslant 0$,

$$Z_t = \int_0^t \int_{\mathbb{R}_+} \mathbb{1}_{[0,\lambda_{s-}]}(x)\Pi(\mathrm{d}s, \mathrm{d}x) \ . \tag{10}$$

Writting $\Pi = \sum_i \delta_{(s_i,x_i)}$, we have $Z_t = \sum_i \mathbb{1}_{s_i \leqslant t}\mathbb{1}_{[0,\lambda_{s_i-}]}(x_i)$: the Poisson point process generates points $(s_i, x_i) \in \mathbb{R}_+ \times \mathbb{R}_+$, and $Z_{s_i} = Z_{s_i-} + 1$ when $\lambda_{s_i-} \geqslant x_i$.

Inhomogeneous Poisson processes. A univariate counting process whose stochastic intensity is a deterministic function is called an *univariate inhomogeneous Poisson process*. A *multivariate inhomogeneous Poisson process* $\mathbf{N}$ has independent coordinates which are inhomogeneous Poisson processes; in particular, its $\mathbf{F}^{\mathbf{N}}$-stochastic intensity is a locally integrable function $t \mapsto \alpha_t = (\alpha_t^1, \ldots, \alpha_t^d) \in \mathbb{R}_+^d$. Alternatively, we may define it through the following transition mechanism.

Lemma 3. *The jump times $(T_i)_{i\geqslant 1}$ of an inhomogeneous Poisson process $\mathbf{N}$ satisfy, for every $i \geqslant 0$,*

$$\mathbb{P}(T_{i+1} > t | \mathcal{F}_{T_i}^{\mathbf{N}}) = \exp\left(-\int_{T_i}^t \|\alpha_s\|_1 \mathrm{d}s\right) \ ,$$

with $T_0 = 0$. Given the $(i+1)$-th jump time, the coordinate that jumps is chosen proportionally to the intensity, namely, for every $1 \leqslant k \leqslant d$,

$$\mathbb{P}(N_{T_{i+1}}^k = N_{T_i}^k + 1 | \mathcal{F}_{T_i}^{\mathbf{N}}, T_{i+1}) = \alpha_{T_{i+1}}^k / \|\alpha_{T_{i+1}}\|_1 \ .$$

Proof. This characterization of an inhomogeneous Poisson process is relatively standard: for completeness, we justify it below.

The sum of $\mathbf{N}$ coordinates, $\|\mathbf{N}\|_1$, is a univariate, inhomogeneous Poisson process with $\mathbf{F}^{\mathbf{N}}$-intensity $t \mapsto \|\alpha_t\|_1$. To see it, we may compute for $t > s$,

$$\mathbb{E}\left[\|N_t\|_1 - \|N_s\|_1 | \mathcal{F}_s^{\mathbf{N}}\right] = \sum_{k=1}^d \mathbb{E}\left[N_t^k - N_s^k | \mathcal{F}_s^{\mathbf{N}}\right] = \int_s^t \|\alpha_u\|_1 \mathrm{d}u \ .$$

Now, using (10), $\|\mathbf{N}\|_1$ can we written as $\|N_t\|_1 = \int_0^t \int_{\mathbb{R}_+} \mathbb{1}_{[0,\|\alpha_s\|_1]}(x)\Pi(\mathrm{d}s, \mathrm{d}x)$, where Π is a $\mathbf{F}^{\mathbf{N}}$-Poisson point process. Then, for $t \geqslant T_i$,

$$\begin{aligned}\mathbb{P}(T_{i+1} > t | \mathcal{F}_{T_i}^{\mathbf{N}}) &= \mathbb{P}\left(\int_{T_i}^t \int_{\mathbb{R}_+} \mathbb{1}_{[0,\|\alpha_s\|_1]}(x)\Pi(\mathrm{d}s, \mathrm{d}x) = 0 \middle| \mathcal{F}_{T_i}^{\mathbf{N}}\right) \\ &= \exp\left(-\int_{T_i}^t \|\alpha_s\|_1 \mathrm{d}s\right) \ ,\end{aligned} \tag{11}$$

as, by definition of the Poisson point process, conditionally on $\mathcal{F}_{T_i}$,

$$\int_{T_i}^t \int_{\mathbb{R}_+} \mathbb{1}_{[0,\|\alpha_s\|_1]}(x)\Pi(\mathrm{d}s, \mathrm{d}x) \sim \mathrm{Pois}\left(\int_{T_i}^t \int_{\mathbb{R}_+} \mathbb{1}_{[0,\|\alpha_s\|_1]}(x)\mathrm{d}s\mathrm{d}x\right) \ ,$$

Here, we used the strong Markov property of the $\mathbf{F}^{\mathbf{N}}$-Poisson point process Π: conditionally on $\mathcal{F}_{T_i}^{\mathbf{N}}$, with T_i a $\mathbf{F}^{\mathbf{N}}$-stopping time, the restriction of Π to $[T_i, +\infty) \times \mathbb{R}_+$ is still a Poisson point process. Similarly, we can obtain the jump distribution of each coordinate process:

$$\begin{aligned}\mathbb{P}(N_t^j = N_{T_i}^j | \mathcal{F}_{T_i}^{\mathbf{N}}) &= \mathbb{P}\left(\int_{T_i}^t \int_{\mathbb{R}_+} \mathbb{1}_{[0,\alpha_s^j]}(x)\Pi^j(\mathrm{d}s, \mathrm{d}x) = 0 \middle| \mathcal{F}_{T_i}^{\mathbf{N}}\right) \\ &= \exp\left(-\int_{T_i}^t \alpha_s^j \mathrm{d}s\right) \ .\end{aligned} \tag{12}$$

We now identify the mark distribution, *i.e.*, conditionally on $\mathcal{F}_{T_i}^{\mathbf{N}}, T_{i+1}$, the probability that $N_{T_{i+1}}^k = N_{T_i}^k + 1$ for a fixed $1 \leqslant k \leqslant d$. Conditionally on $\mathcal{F}_{T_i}^{\mathbf{N}}$, define

$$t \geqslant T_i \mapsto p_j(t) = \alpha_t^j \exp\left(-\int_{T_i}^t \alpha_s^j \mathrm{d}s\right) \ , 1 \leqslant j \leqslant d \ ,$$

and $t \geqslant T_i \mapsto p(t) = \|\alpha_t\|_1 \exp\left(-\int_{T_i}^t \|\alpha_s\|_1 \mathrm{d}s\right)$. From (12),(11), we recognize the conditional density of the first jump time of $\mathbf{N}^j$ (respectively $\|\mathbf{N}\|_1$) on $[T_i, +\infty)$. Then, we compute the probability that $\mathbf{N}^k$ is the first coordinate to jump after T_i, and do so before a given $t \geqslant T_i$:

$$\begin{aligned}\mathbb{P}(T_{i+1} \leqslant t, N^k_{T_{i+1}} = N^k_{T_i} + 1 | \mathcal{F}^{\mathbf{N}}_{T_i}) &= \int_{T_i}^t p_k(s_k) \prod_{j \neq k} \int_{s_k}^\infty p_j(s_j) \mathrm{d}s_j \mathrm{d}s_k \\ &= \int_{T_i}^t p_k(s_k) \prod_{j \neq k} \exp\left(-\int_{T_i}^{s_k} \alpha^j_u \mathrm{d}u\right) \mathrm{d}s_k \\ &= \int_{T_i}^t \alpha^k_{s_k} \exp\left(-\int_{T_i}^{s_k} \|\alpha_u\|_1 \mathrm{d}u\right) \mathrm{d}s_k \ .\end{aligned}$$

Here, we used the fact that the coordinate processes are independent. We may also compute this probability by considering that the next jump of $\|\mathbf{N}\|_1$ occurs at time s, and introduce the probability $q_k(s)$ that this jump actually occurs on coordinate k:

$$\begin{aligned}\mathbb{P}(T_{i+1} \leqslant t, N^k_{T_{i+1}} = N^k_{T_i} + 1 | \mathcal{F}^{\mathbf{N}}_{T_i}) &= \int_{T_i}^t p(s) q_k(s) \mathrm{d}s \\ &= \int_{T_i}^t \|\alpha_s\|_1 \exp\left(-\int_{T_i}^s \|\alpha_u\|_1 \mathrm{d}u\right) q_k(s) \mathrm{d}s \ .\end{aligned}$$

These two expressions coincide by setting $q_k(s) = \alpha^k_s / \|\alpha_s\|_1$, yielding the desired mark distribution. □

Consider now a multivariate counting process $\mathbf{Z}$ whose $\mathbf{F}^{\mathbf{Z}}$-stochastic intensity is of the form $t \mapsto \lambda_t = u_t(Z_{t-})$, where, for any $z \in \mathbb{N}^d$, $t \mapsto u_t(z)$ is locally integrable. Observe that between two consecutive jumps, *i.e.*, on $[T_i, T_{i+1})$, conditionally on $\mathcal{F}_{T_i}$, $\mathbf{Z}$ coincides with an inhomogeneous Poisson process with intensity deterministic $t \mapsto u_t(Z_{T_i})$. As a result, we can extend Lemma 3 as follow.

Proposition 4. *The jump times $(T_i)_{i \geqslant 1}$ of a counting process $\mathbf{Z}$ with intensity of the form $t \mapsto u_t(Z_{t-})$ satisfy, for every $i \geqslant 0$,*

$$\mathbb{P}(T_{i+1} > t | \mathcal{F}^{\mathbf{Z}}_{T_i}) = \exp\left(-\int_{T_i}^t \|u_s(Z_{T_i})\|_1 \mathrm{d}s\right) \ ,$$

with $T_0 = 0$. Given the $(i+1)$-th jump time, the coordinate that jumps is chosen proportionally to the intensity, namely, for every $1 \leqslant k \leqslant d$,

$$\mathbb{P}(Z^k_{T_{i+1}} = Z^k_{T_i} + 1 | \mathcal{F}^{\mathbf{Z}}_{T_i}, T_{i+1}) = u^k_{T_{i+1}}(Z_{T_i}) / \|u_{T_{i+1}}(Z_{T_i})\|_1 \ .$$

In particular, $\mathbf{Z}$ is a Markov process.

B.2 MPP PROPERTIES: PROOFS OF SECTION 2.1

Proposition 2. *Given any $\mathbf{Z} \sim \mathrm{MPP}(\pi)$, the Poisson denoiser K^π defined by* (1) *can be expressed as follows. For every $t > 0$ and $z \in \mathbb{N}^d$,*

$$K^\pi_t(z) = \int_{\mathbb{R}^d_+} x \prod_{i=1}^d \mathrm{e}^{-t x_i} x_i^{z_i} \pi(\mathrm{d}x) \Big/ \int_{\mathbb{R}^d_+} \prod_{i=1}^d \mathrm{e}^{-t x_i} x_i^{z_i} \pi(\mathrm{d}x) \tag{3}$$

holds if $\mathbb{P}(Z_t = z) > 0$; otherwise, by convention, $K^\pi_t(z) = 0$. Furthermore, for any $z \in \mathbb{N}^d$, $t > 0 \mapsto K^\pi_t(z)$ is smooth and $t > 0 \mapsto \|K^\pi_t(z)\|_1$ is non-increasing.

Proof. This expression of the denoiser is derived in (Zocher, 2005, Theorem 3.4.2, 4.4.2); we include it here for completeness. Let $\mathbf{Z} \sim \mathrm{MPP}(\pi)$, with random rate X. Fix $t > 0, z \in \mathbb{N}^d$. Since Z_t, conditionally on X, has independent coordinates distributed as $\mathcal{P}(tX_i), 1 \leqslant i \leqslant d$, we have

$$\mathbb{P}(Z_t = z) = \int_{\mathbb{R}^d_+} p_t(z|x) \pi(\mathrm{d}x) \ , \quad p_t(z|x) \coloneqq \prod_{j=1}^d \mathrm{e}^{-t x_j} \frac{(t x_j)^{z_j}}{z_j!} \ . \tag{13}$$

This probability may vanish if, e.g., $\pi(\{0\} \times \mathbb{R}_+^{d-1}) = 1$ and $z_1 \geqslant 1$; it is non-zero if and only if $\pi((\{z_1\} \cup (0, +\infty)) \times \cdots \times (\{z_d\} \cup (0, +\infty))) > 0$, which we assume from now on.

Using Bayes' rule leads to the conditional distribution of X given $Z_t = z$, $\pi_t(\mathrm{d}x|z) = p_t(z|x)\pi(\mathrm{d}x)/\mathbb{P}(Z_t = z)$; hence for any $1 \leqslant i \leqslant d$, the denoiser i-th coordinate satisfies

$$\begin{aligned}
[K_t^\pi(z)]_i &= \mathbb{E}[X_i | Z_t = z] \\
&= \frac{1}{\mathbb{P}(Z_t = z)} \int_{\mathbb{R}_+^d} x_i \times p_t(z|x)\pi(\mathrm{d}x) \\
&= \frac{1}{\mathbb{P}(Z_t = z)} \int_{\mathbb{R}_+^d} \frac{z_i + 1}{t} \frac{(tx_i)^{z_i+1}}{(z_i+1)!} \times \prod_{j \neq i} \mathrm{e}^{-tx_j} \frac{(tx_j)^{z_j}}{z_j!} \pi(\mathrm{d}x) \\
&= \frac{z_i + 1}{t} \frac{\mathbb{P}(Z_t = z + e_i)}{\mathbb{P}(Z_t = z)} \ ,
\end{aligned}$$

recognizing (13) with $z + e_i$. Using again (13), we get

$$\begin{aligned}
[K_t^\pi(z)]_i &= \int_{\mathbb{R}_+^d} x_i \prod_{j=1}^d \mathrm{e}^{-tx_j} \frac{(tx_j)^{z_j}}{z_j!} \pi(\mathrm{d}x) \Bigg/ \int_{\mathbb{R}_+^d} \prod_{j=1}^d \mathrm{e}^{-tx_j} \frac{(tx_j)^{z_j}}{z_j!} \pi(\mathrm{d}x) \\
&= \int_{\mathbb{R}_+^d} x_i \prod_{j=1}^d \mathrm{e}^{-tx_j} x_j^{z_j} \pi(\mathrm{d}x) \Bigg/ \int_{\mathbb{R}_+^d} \prod_{j=1}^d \mathrm{e}^{-tx_j} x_j^{z_j} \pi(\mathrm{d}x) \ ,
\end{aligned}$$

after simplifying the common factors, which proves the first part of the proposition.

Smoothness of $t > 0 \mapsto K_t^\pi(z)$ follows from the smoothness of the Laplace transform of π; we refer to (Zocher, 2005, Theorem 3.4.5) for details.

Regarding the last point of the proposition let $S \coloneqq \sum_{j=1}^d X_j$, $P \coloneqq \prod_{j=1}^d X_j^{z_j}$; then for $t > 0$,

$$\|K_t^\pi(z)\|_1 = \sum_{i=1}^d \mathbb{E}[X_i \mathrm{e}^{-tS} P] / \mathbb{E}[\mathrm{e}^{-tS} P] = \mathbb{E}[S \mathrm{e}^{-tS} P] / \mathbb{E}[\mathrm{e}^{-tS} P] \ ,$$

$$\mathbb{E}[\mathrm{e}^{-tS} P]^2 \times \frac{\mathrm{d}}{\mathrm{d}t} \|K_t^\pi(z)\|_1 = \mathbb{E}[S \mathrm{e}^{-tS} P] \mathbb{E}[S \mathrm{e}^{-tS} P] - \mathbb{E}[S^2 \mathrm{e}^{-tS} P] \mathbb{E}[\mathrm{e}^{-tS} P] \ . \tag{14}$$

From the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\mathbb{E}[S \mathrm{e}^{-tS} P]^2 &= \mathbb{E}[S \mathrm{e}^{-tS/2} P^{1/2} \times \mathrm{e}^{-tS/2} P^{1/2}]^2 \\
&\leqslant \mathbb{E}[S^2 \mathrm{e}^{-tS} P] \mathbb{E}[\mathrm{e}^{-tS} P] \ ,
\end{aligned}$$

thus (14) $\leqslant 0$, which concludes the proof. □

Theorem 1. *Let* $\mathbf{Z} \sim \mathrm{MPP}(\pi)$*, with random rate* X*. Almost surely,* $\lim_{t \to \infty} Z_t/t = X$.

Proof. The proof is a straightforward generalization of the univariate case (Grandell, 1997, Proposition 4.2). Conditioning on X, we get

$$\mathbb{P}(Z_t/t \to X) = \mathbb{E}\left[\mathbb{P}\left(\bigcap_{i=1}^d \{Z_t^i/t \to X_i\} \right) \middle| X \right] \ .$$

By definition, conditionally on X, the coordinates $\mathbf{Z}^i, 1 \leqslant i \leqslant d$, are Poisson processes with respective intensities $X_i, 1 \leqslant i \leqslant d$. As a consequence, it suffices to check that for every $x > 0, N_t^x/t \to x$, where $\mathbf{N}^x$ is a univariate Poisson process with rate x, whose jump times are denoted by $(T_n^x)_{n \geqslant 1}$. The case $x = 0$ is obvious. If $x > 0$, the strong law of large numbers gives $T_n^x/n \to 1/x$ a.s., since the increments $T_n^x - T_{n-1}^x$ are independent and distributed as $\mathrm{Exp}(x)$; in particular, $T_n^x \to +\infty$ a.s. Now fix $\varepsilon > 0$: almost surely, there exist $p \in \mathbb{N}$ such that $x - \varepsilon \leqslant n/T_{n+1}$ and $n/T_n \leqslant x + \varepsilon$ for any $n \geqslant p$. If $t \geqslant T_p^x$, there exists $n \geqslant p$ satisfying $T_n^x \leqslant t < T_{n+1}^x$, and

$$x - \varepsilon \leqslant n/T_{n+1} \leqslant N_t^x/t \leqslant n/T_n \leqslant x + \varepsilon \ ,$$

concluding the proof. □

Theorem 2. *The $\mathbf{F}^{\mathbf{Z}}$-compensator of a mixed Poisson process $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ is the process defined for every $t \geqslant 0$ by*

$$\Lambda_t = \int_0^t K_s^\pi(Z_s)\mathrm{d}s \ .$$

In other words, $\mathbf{Z}$ is a Markov counting process with $\mathbf{F}^{\mathbf{Z}}$-stochastic intensity $t \in \mathbb{R}_+ \mapsto K_t^\pi(Z_{t-})$.

Proof. Let $X \sim \pi$ denote the random rate of $\mathbf{Z}$. First, we show that the result holds by replacing $K_t^\pi(Z_s) = \mathbb{E}[X|Z_s]$ with $\mathbb{E}[X|\mathcal{F}_s^{\mathbf{Z}}]$. Consider the augmented σ-algebra $\mathcal{G}_t = \sigma(X) \vee \mathcal{F}_t^{\mathbf{Z}}$, satisfying $\mathcal{F}_t^{\mathbf{Z}} \subseteq \mathcal{G}_t$. $\mathbf{Z}$ is, conditionally on X, a multivariate Poisson process with intensity X, thus have Poisson-distributed independent increments. Then, for every $t > s \geqslant 0$,

$$\begin{aligned}
\mathbb{E}\left[Z_t - \int_0^t \mathbb{E}[X|\mathcal{F}_u^{\mathbf{Z}}]\mathrm{d}u \middle| \mathcal{F}_s^{\mathbf{Z}}\right] &= Z_s + \mathbb{E}[\mathbb{E}[Z_t - Z_s|\mathcal{G}_s]|\mathcal{F}_s^{\mathbf{Z}}] - \int_0^s \mathbb{E}[X|\mathcal{F}_u^{\mathbf{Z}}]\mathrm{d}u - \int_s^t \mathbb{E}[X|\mathcal{F}_s^{\mathbf{Z}}]du \\
&= Z_s + \mathbb{E}[(t-s)X|\mathcal{F}_s^{\mathbf{Z}}] - \int_0^s \mathbb{E}[X|\mathcal{F}_u^{\mathbf{Z}}]\mathrm{d}u - (t-s)\mathbb{E}[X|\mathcal{F}_s^{\mathbf{Z}}] \\
&= Z_s - \int_0^s \mathbb{E}[X|\mathcal{F}_u^{\mathbf{Z}}]\mathrm{d}u \ ,
\end{aligned}$$

where we used $\mathbb{E}[\mathbb{E}[X|\mathcal{F}_u^{\mathbf{Z}}]|\mathcal{F}_s^{\mathbf{Z}}] = \mathbb{E}[X|\mathcal{F}_s^{\mathbf{Z}}]$ for $u \geqslant s$.

Finally, the structure of the mixed Poisson process ensures that $\mathbb{E}[X|\mathcal{F}_t^{\mathbf{Z}}] = \mathbb{E}[X|Z_t]$. Take $\mathsf{A} = \{Z_{t_1} = z_1, ..., Z_{t_n} = z_n\} \in \mathcal{F}_t^{\mathbf{Z}}$, with $0 = t_0 \leqslant t_1 \leqslant ... \leqslant t_n = t$ and for any $0 \leqslant i \leqslant n, z_i = (z_i^1, \ldots, z_i^d) \in \mathbb{N}^d$ with for every $1 \leqslant j \leqslant d$, $0 = z_0^j \leqslant z_1^j \leqslant ... \leqslant z_n^j$. As, conditionally on X, $\mathbf{Z}$ is a multivariate Poisson process with intensity X, its coordinates are independent univariate Poisson processes, whose increments are independent and Poisson distributed. Consequently,

$$\begin{aligned}
\mathbb{P}(\mathsf{A}|X) &= \mathbb{P}\left(\bigcap_{j=1}^d \bigcap_{i=1}^n \left\{Z_{t_i}^j - Z_{t_{i-1}}^j = z_i^j - z_{i-1}^j\right\} \middle| X\right) \\
&= \prod_{j=1}^d \prod_{i=1}^n \mathrm{e}^{-(t_i - t_{i-1})X_j} \frac{(t_i - t_{i-1})^{z_i^j - z_{i-1}^j} X_j^{z_i^j - z_{i-1}^j}}{(z_i^j - z_{i-1}^j)!} \\
&= \mathrm{e}^{-t\|X\|_1} P \times \prod_{j=1}^d \prod_{i=1}^n \frac{(t_i - t_{i-1})^{z_i^j - z_{i-1}^j}}{(z_i^j - z_{i-1}^j)!} \ ,
\end{aligned}$$

where $P \coloneqq \prod_{j=1}^d X_j^{z_n^j}$. The double product is a constant that we denote C_{A}. Then,

$$\begin{aligned}
\mathbb{E}[\mathbb{1}_{\mathsf{A}} X] &= \mathbb{E}[X\mathbb{P}(\mathsf{A}|X)] \\
&= \mathbb{E}[X\mathrm{e}^{-t\|X\|_1} P] \times C_A \ .
\end{aligned}$$

Furthermore, as $\{Z_t = z_n\} \supseteq \mathsf{A}$ and $\mathbb{E}[X|Z_t]\mathbb{1}_{Z_t = z_n} = \mathbb{E}[X|Z_t = z_n]\mathbb{1}_{Z_t = z_n}$,

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[X|Z_t]\mathbb{1}_{\mathsf{A}}] &= \mathbb{E}[\mathbb{E}[X|Z_t]\mathbb{1}_{Z_t = z_n}\mathbb{1}_{\mathsf{A}}] \\
&= \mathbb{E}[X|Z_t = z_n]\mathbb{P}(\mathsf{A}) \\
&= K_t^\pi(z_n) \times \mathbb{E}[\mathrm{e}^{-t\|X\|_1} P] \times C_{\mathsf{A}} \\
&= \mathbb{E}[X\mathrm{e}^{-t\|X\|_1} P] \times C_{\mathsf{A}} \ ,
\end{aligned}$$

where we use Proposition 2 to obtain the last line. Thus $\mathbb{E}[\mathbb{E}[X|Z_t]\mathbb{1}_{\mathsf{A}}] = \mathbb{E}[\mathbb{1}_{\mathsf{A}} X]$ which shows that $\mathbb{E}[X|Z_t] = \mathbb{E}[X|\mathcal{F}_t^{\mathbf{Z}}]$.

Because its stochastic intensity at t is Z_t-measurable rather than simply $\mathcal{F}_t^{\mathbf{Z}}$-measurable, it is relatively straightforward to see that $\mathbf{Z}$ is a Markov process; see Proposition 4. Alternatively, we can prove it directly as follows. Keep the introduced notation, and let $\mathsf{A}' = \{Z_{t_1} = z_1, \ldots, Z_{t_{n-1}} = z_{n-1}\} \in \mathcal{F}_{t_{n-1}}^{\mathbf{Z}}$; we have $\mathsf{A} \subseteq \mathsf{A}'$. Notice that

$$C_{\mathsf{A}}/C_{\mathsf{A}'} = \prod_{j=1}^d (t - t_{n-1})^{z_n^j - z_{n-1}^j}/(z_n^j - z_{n-1}^j)! \ .$$

Then,

$$\mathbb{P}(Z_t = z_n|\mathsf{A}') = \mathbb{P}(\mathsf{A})/\mathbb{P}(\mathsf{A}')$$
$$= \mathbb{E}\left[\mathrm{e}^{-t\|X\|_1}\prod_{j=1}^{d} X_j^{z_n^j}\right] \Big/ \mathbb{E}\left[\mathrm{e}^{-t_{n-1}\|X\|_1}\prod_{j=1}^{d} X_j^{z_{n-1}^j}\right] \times \prod_{j=1}^{d}\frac{(t-t_{n-1})^{z_n^j - z_{n-1}^j}}{(z_n^j - z_{n-1}^j)!} .$$

This expression does not depend on $t_i, z_i, i < n-1$: the same computation, but replacing A' with $\{Z_{t_{n-1}} = z_{n-1}\}$, yields the same expression: thus $\mathbb{P}(Z_t = z_n|\mathsf{A}') = \mathbb{P}(Z_t = z_n|Z_{t_{n-1}} = z_{n-1})$, hence the Markov property. □

B.3 Simulation algorithms: proofs and illustrations

Proposition 5. *Consider a counting process $\mathbf{Z}$ whose $\mathbf{F}^{\mathbf{Z}}$-intensity has the form $t \mapsto \lambda_t = u_t(Z_{t-}) \in \mathbb{R}_+^d$, with u a locally integrable function such that for every $z \in \mathbb{N}^d$, $t \mapsto \|u_t(z)\|_1$ is positive and non-increasing. Let $\bar{\mathbf{Z}}$ denote the counting process recursively defined as follows.*

Set $\bar{T}_0 = 0$. For every $i \geqslant 0$, conditionally on $\bar{T}_i, \bar{Z}_{\bar{T}_i}$, draw the jump times of a univariate Poisson process on $[\bar{T}_i, +\infty)$ with constant intensity $\|u_{\bar{T}_i}(\bar{Z}_{\bar{T}_i})\|_1$, that is, $\bar{T}_i < S_1 < S_2 < \dots$ with iid increments distributed as $\mathrm{Exp}(\|u_{\bar{T}_i}(\bar{Z}_{\bar{T}_i})\|_1)$.

Starting from $k = 1$, draw $U_k \sim \mathcal{U}_{[0,1]}$. If $U_k < \|u_{S_k}(\bar{Z}_{\bar{T}_i})\|_1/\|u_{\bar{T}_i}(\bar{Z}_{\bar{T}_i})\|_1$, set $\bar{T}_{i+1} = S_k$, draw $j \sim \mathrm{Cat}(u_{\bar{T}_{i+1}}(\bar{Z}_{\bar{T}_i}))$ and set $\bar{Z}_{\bar{T}_{i+1}} = \bar{Z}_{\bar{T}_i} + e_j$, $e_j = (\delta_{j,l})_{l=1}^d$. Else, increment k and repeat.

Then, both processes $\bar{\mathbf{Z}}, \mathbf{Z}$ have the same distribution.

Proof. As, conditionally on $\mathbf{F}^{\mathbf{Z}}_{T_i}$, the counting process $\mathbf{Z}$ coincides with an inhomogeneous Poisson process with deterministic intensity $t \mapsto u_t(Z_{T_i}) =: \alpha_t$ until its next jump, Proposition 4 gives

$$\mathbb{P}(T_{i+1} > t|Z_{T_i}, T_i) = \exp\left(-\int_{T_i}^{t}\|\alpha_s\|_1 \mathrm{d}s\right) ,$$

and coordinate j is chosen with probability $\alpha^j_{T_{i+1}}/\|\alpha_{T_{i+1}}\|_1$, *i.e.*, the same mechanism used to simulate $\bar{\mathbf{Z}}$.

Now, we must verify that the distribution of $(\bar{T}_i)_{i\geqslant 0}$ coincides with that of $(T_i)_{i\geqslant 0}$. Fix $i \in \mathbb{N}$ and $t \geqslant \bar{T}_i$. Conditionally on $\bar{T}_i, \bar{Z}_{\bar{T}_i}$, let M denote the number of jumps of a univariate Poisson process on $[\bar{T}_i, t]$ with constant intensity $\|u_{\bar{T}_i}(\bar{Z}_{\bar{T}_i})\|_1 = \|\alpha_{\bar{T}_i}\|_1$, and $\bar{T}_i < S_1 < \dots < S_M$ the associated jump times. Conditionally on $\bar{T}_i, \bar{Z}_{\bar{T}_i}$ and M, which follows a Poisson distribution with mean $(t-\bar{T}_i)\|\alpha_{\bar{T}_i}\|_1$, the unordered jump times are independent and uniformly distributed on $[\bar{T}_i, t]$. Take $U_1, U_2, ...$ a sequence of iid random variables, uniform on $[0,1]$, independent of the other random variables. Then, no jump before t means that every proposed jump is rejected, that is,

$$\mathbb{P}(\bar{T}_{i+1} > t|\bar{Z}_{\bar{T}_i}, \bar{T}_i) = \sum_{n\geqslant 0}\mathbb{P}(M = n|\bar{Z}_{\bar{T}_i}, \bar{T}_i)\mathbb{P}\left(\bigcap_{k=1}^{n}\{U_k > \|\alpha_{S_k}\|_1/\|\alpha_{\bar{T}_i}\|_1\}\middle| M = n, \bar{Z}_{\bar{T}_i}, \bar{T}_i\right)$$

$$= \exp\left(-(t-\bar{T}_i)\|\alpha_{\bar{T}_i}\|_1\right)\sum_{n\geqslant 0}\frac{((t-\bar{T}_i)\|\alpha_{\bar{T}_i}\|_1)^n}{n!}\left(\frac{1}{t-\bar{T}_i}\int_{\bar{T}_i}^{t} 1 - \frac{\|\alpha_s\|_1}{\|\alpha_{\bar{T}_i}\|_1}\mathrm{d}s\right)^n$$

$$= \exp\left(-(t-\bar{T}_i)\|\alpha_{\bar{T}_i}\|_1\right)\exp\left((t-\bar{T}_i)\|\alpha_{\bar{T}_i}\|_1 - \int_{\bar{T}_i}^{t}\|\alpha_s\|_1\mathrm{d}s\right)$$

$$= \exp\left(-\int_{\bar{T}_i}^{t}\|\alpha_s\|_1\mathrm{d}s\right) ,$$

which equals $\mathbb{P}(T_{i+1} > t|Z_{T_i}, T_i)$. Both $\mathbf{Z}$ and $\bar{\mathbf{Z}}$ are Markovian, and their conditional distributions of the next jump are identical; thus, their distributions match. □

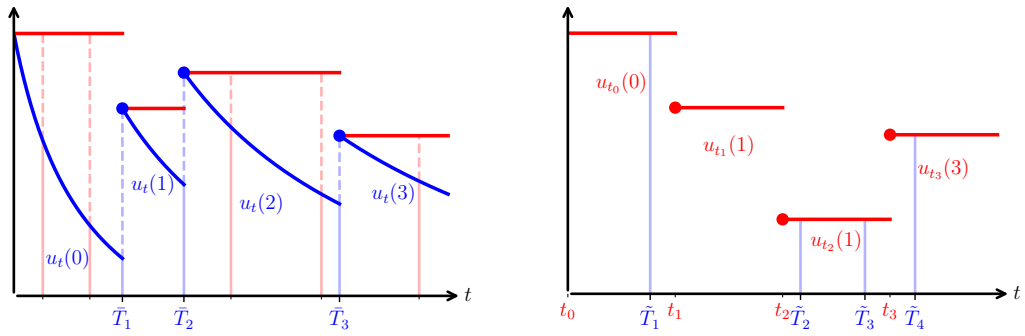

Figure 2: Visual representation of the simulation procedures when π is a Gamma distribution with paramaters $\alpha, \beta = 1, 2$. **(Left)** The thinning algorithm of Proposition 5, Algorithm 2, in dimension $d = 1$. For every $i \geqslant 0, t \geqslant \bar{T}_i$, potential jump times (vertical lines) are drawn according to the "frozen", piecewise constant intensity $u_{\bar{T}_i}(i)$ (red curve) instead of the true intensity $t \mapsto u_t(i)$ (blue curve); and accepted with probability $u_t(i)/u_{\bar{T}_i}(i) = (- - -)/(\text{———}+ - - -)$. Accepted jumps times $\bar{T}_i$ are colored blue. **(Right)** The approximate process of Proposition 6, Algorithm 1. Given a time-grid $0 = t_0 < t_1 < t_2 < \dots$, jump times $\tilde{T}_i$ are drawn according to a discretized version of the intensity (red curve): on each sub-interval $[t_k, t_{k+1}]$, the number of jumps is distributed as $\text{Pois}((t_{k+1} - t_k)u_{t_k}(\tilde{Z}_{t_k}))$.

Algorithm 2 Thinning

Require: Intensity $(t, z) \mapsto u_t(z) \in \mathbb{R}_+^d$ satisfying the assumptions of Proposition 5; $t_{\max} > 0$

Initialize $i = 0$, $t_0 = 0$, $s = 0$ and $\bar{Z}_0 = 0 \in \mathbb{N}^d$
repeat
  Increment $s = s + \text{Exp}(\|u_{t_i}(\bar{Z}_i)\|_1)$ ▷ Potential next jump time
  Compute $u_s(\bar{Z}_i)$
  Draw $U \sim \text{Unif}_{[0,1]}$
  if $U < \|u_s(\bar{Z}_i)\|_1/\|u_{t_i}(\bar{Z}_i)\|_1$ and $s < t_{\max}$ **then** ▷ Accept or reject jump
    Draw $j \sim \text{Cat}(u_s(\bar{Z}_i))$ ▷ If accepted, choose which coordinate actually jumps
    Set $\bar{Z}_{i+1} = \bar{Z}_i + e_j, t_{i+1} = s$
    Increment $i = i + 1$
  end if
until $s \geqslant t_{\max}$
return $\bar{Z}_i$

Proposition 6. *Let $\mathbf{Z}$ be a counting process with $\mathbf{F}^{\mathbf{Z}}$-intensity $t \mapsto u_t(Z_{t-}) \in \mathbb{R}_+^d$, where u is a measurable, locally integrable function. For any $t > 0$, consider a time-grid $0 = t_0 < t_1 < \cdots < t_N = t$, with $\Delta_k := t_{k+1} - t_k$. Define the Euler discretization $\tilde{\mathbf{Z}}$ of $\mathbf{Z}$ as the counting process with the following recursion: $\tilde{Z}_0 = 0$ and for every $k \geqslant 0, s \in [t_k, t_{k+1}]$, conditionally on $\tilde{Z}_{t_k}$,*

$$\tilde{Z}_s = \tilde{Z}_k + {}^k N_{s-t_k}^{u_{t_k}(\tilde{Z}_{t_k})} ,$$

where the ${}^k\mathbf{N}^{u_{t_k}(\tilde{Z}_{t_k})}$ is a standard Poisson processes with constant intensity $u_{t_k}(\tilde{Z}_{t_k})$ and independent of the past. Said differently, $\tilde{\mathbf{Z}}$ has piecewise constant $\mathbf{F}^{\tilde{\mathbf{Z}}}$-stochastic intensity $s \mapsto \lambda_s := \sum_{k=0}^{N-1} \mathbb{1}_{[t_k, t_{k+1})}(s) u_{t_k}(\tilde{Z}_{t_k})$. Assume the following.

- *Given any fixed $z \in \mathbb{N}^d$, $s \in [0, t] \mapsto u_s(z)$ is Lipschitz continuous, and there exist constants $a, b, L \geqslant 0$, as well as $0 \leqslant L_k \leqslant L, 0 \leqslant k \leqslant N - 1$, such that the Lipschitz constant of $s \in [t_k, t_{k+1}] \mapsto u_s(z)$ is upper bounded by $(a\|z\|_1 + b)L_k$.*
- *For a fixed $s \in [0, t]$, $\text{Lip}(u_s) := \sup_{z \neq y \in \mathbb{N}^d} \|u_s(z) - u_s(y)\|_1/\|z - y\|_1$ if finite, and $\sup_{s \leqslant t} \text{Lip}(u_s) < \infty$. This holds in particular if $s \mapsto \text{Lip}(u_s)$ is continuous.*

Then, $\mathbb{E}[\|\tilde{Z}_t - Z_t\|_1] = \mathcal{O}(\max_k \Delta_k)$ *as* $\max_k \Delta_k \to 0$.

In particular, these assumptions hold when $\mathbf{Z} \sim \mathrm{MPP}(\gamma)$ with $\gamma \sim \mathbf{\Gamma}(\alpha, \beta)$ a Gamma distribution, in which case $u_t^i(z) = (z_i + \alpha_i)/(\beta_i + t)$; see Appendix B.4.

Proof. First, we derive an upper bound on the expected values of $\|\tilde{\mathbf{Z}}\|_1$ on the grid. Write, for $s \in [t_k, t_{k+1}]$,

$$\begin{aligned}
\mathbb{E}[\|\tilde{Z}_s\|_1] &= \mathbb{E}[\|\tilde{Z}_{t_k}\|_1] + \mathbb{E}[\|^k N_{s-t_k}^{u_{t_k}(\tilde{Z}_{t_k})}\|_1] \\
&= \mathbb{E}[\|\tilde{Z}_{t_k}\|_1] + (s - t_k)\mathbb{E}[\|u_{t_k}(\tilde{Z}_{t_k})\|_1] \\
&\leqslant \mathbb{E}[\|\tilde{Z}_{t_k}\|_1] + (s - t_k)\left(\|u_{t_k}(0)\|_1 + \mathrm{Lip}(u_{t_k})\mathbb{E}[\|\tilde{Z}_{t_k}\|_1]\right) , \qquad (15)
\end{aligned}$$

and in particular $\mathbb{E}[\|\tilde{Z}_{t_{k+1}}\|_1] \leqslant \Delta_k\|u_{t_k}(0)\|_1 + \mathbb{E}[\|\tilde{Z}_{t_k}\|](1 + \Delta_k \mathrm{Lip}(u_{t_k}))$; writing $\mu_k := \mathbb{E}[\|\tilde{Z}_{t_k}\|_1]$, increasing in k due to counting processes being non-decreasing, this gives the upper bound

$$\begin{aligned}
\mu_k \leqslant \mu_N &\leqslant \sum_{i=0}^{N-1} \Delta_i \|u_{t_i}(0)\|_1 \prod_{j=i+1}^{N-1} (1 + \Delta_j \mathrm{Lip}(u_{t_j})) \\
&\leqslant \sum_{i=0}^{N-1} \Delta_i \|u_{t_i}(0)\|_1 \times \exp\left(\sum_{j=0}^{N-1} \Delta_j \mathrm{Lip}(u_{t_j})\right) \\
&\leqslant \sum_{i=0}^{N-1} \Delta_i \|u_{t_i}(0)\|_1 \times \exp\left(t \sup_{s \leqslant t} \mathrm{Lip}(u_s)\right) \\
&\leqslant t \sup_{s \leqslant t} \|u_s(0)\|_1 \times \exp\left(t \sup_{s \leqslant t} \mathrm{Lip}(u_s)\right) .
\end{aligned}$$

In particular, there exists a uniform upper bound $K > 0$ such that $0 = \mu_0 \leqslant \mu_1 \leqslant \cdots \leqslant \mu_N \leqslant K$ for any time-grid.

Next, we use the integral representation (10) with $\mathbf{F} = \mathbf{F}^{\mathbf{Z}} \vee \mathbf{F}^{\tilde{\mathbf{Z}}}$ and couples both processes through the same $\mathbf{F}$-Poisson point process Π. For every $1 \leqslant i \leqslant d$,

$$\begin{aligned}
\mathbb{E}[|Z_t^i - \tilde{Z}_t^i|] &\leqslant \mathbb{E}\int_0^t \int_{\mathbb{R}_+} |\mathbb{1}_{[0, u_s^i(Z_{s-})]}(x) - \mathbb{1}_{[0, \lambda_{s-}^i]}(x)| \Pi(\mathrm{d}s, \mathrm{d}x) \\
&= \mathbb{E}\int_0^t \int_{\mathbb{R}_+} |\mathbb{1}_{[0, u_s^i(Z_s)]}(x) - \mathbb{1}_{[0, \lambda_s^i]}(x)| \mathrm{d}x\mathrm{d}s \\
&= \mathbb{E}\int_0^t |u_s^i(Z_s) - \lambda_s^i| \mathrm{d}s \\
&= \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[|u_s^i(Z_s) - u_{t_k}^i(\tilde{Z}_{t_k})|] \mathrm{d}s \\
&\leqslant \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[|u_s^i(\tilde{Z}_s) - u_{t_k}^i(\tilde{Z}_{t_k})|] \mathrm{d}s + \int_0^t \mathbb{E}[|u_s^i(Z_s) - u_s^i(\tilde{Z}_s)|] \mathrm{d}s ,
\end{aligned}$$

where the second line is obtained by noting that $s \mapsto |\mathbb{1}_{[0, u_s^i(Z_{s-})]}(x) - \mathbb{1}_{[0, \lambda_{s-}^i]}(x)|$ is $\mathcal{F}_s$-predictable by definition of stochastic intensities, and using the compensation formula of the Poisson random measure; see, e.g., (Kyprianou, 2014, Theorem 4.4). Note that we omit the $s-$ in the integral, as, almost surely, this does not change the integral value. Summing these inequalities, we obtain

$$\mathbb{E}[\|Z_t - \tilde{Z}_t\|_1] \leqslant \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|u_s(\tilde{Z}_s) - u_{t_k}(\tilde{Z}_{t_k})\|_1] \mathrm{d}s + \int_0^t \mathrm{Lip}(u_s)\mathbb{E}[\|Z_s - \tilde{Z}_s\|_1] \mathrm{d}s .$$

To bound the first term, write, for $s \in [t_k, t_{k+1}]$,

$$\begin{aligned}\mathbb{E}[\|u_s(\tilde{Z}_s) - u_{t_k}(\tilde{Z}_{t_k})\|_1] &\leqslant \mathbb{E}[\|u_s(\tilde{Z}_s) - u_s(\tilde{Z}_{t_k})\|_1] + \mathbb{E}[\|u_s(\tilde{Z}_{t_k}) - u_{t_k}(\tilde{Z}_{t_k})\|_1] \\ &\leqslant \mathrm{Lip}(u_s)\mathbb{E}[\|\tilde{Z}_s - \tilde{Z}_{t_k}\|_1] + (s-t_k)L_k(a\mathbb{E}[\|\tilde{Z}_{t_k}\|_1] + b) \\ &\leqslant \mathrm{Lip}(u_s)(s-t_k)(\|u_{t_k}(0)\|_1 + \mathrm{Lip}(u_{t_k})\mathbb{E}[\|\tilde{Z}_{t_k}\|_1]) + (s-t_k)L_k(a\mathbb{E}[\|\tilde{Z}_{t_k}\|_1] + b) \\ &= \mathrm{Lip}(u_s)(s-t_k)(\|u_{t_k}(0)\|_1 + \mathrm{Lip}(u_{t_k})\mu_k) + (s-t_k)L_k(a\mu_k + b)\ ,\end{aligned}$$

using (15) and the assumptions on u. This yields

$$\begin{aligned}&\sum_{k=0}^{N-1}\int_{t_k}^{t_{k+1}} \mathbb{E}[\|u_s(\tilde{Z}_s) - u_{t_k}(\tilde{Z}_{t_k})\|_1]\mathrm{d}s \\ &\leqslant \sum_{k=0}^{N-1}(\|u_{t_k}(0)\|_1 + \mathrm{Lip}(u_{t_k})\mu_k)\int_{t_k}^{t_{k+1}} \mathrm{Lip}(u_s)(s-t_k)\mathrm{d}s + \sum_{k=0}^{N-1}\frac{\Delta_k^2}{2}L_k(a\mu_k+b) \qquad (16) \\ &\leqslant C \coloneqq \sum_{k=0}^{N-1}\left(\sup_{s\leqslant t}\mathrm{Lip}(u_s)(\sup_{s\leqslant t}\|u_s(0)\|_1 + \sup_{s\leqslant t}\mathrm{Lip}(u_s)K) + L(aK+b)\right)\frac{\Delta_k^2}{2}\ .\end{aligned}$$

Using Gronwall's lemma, the discretization error upper bound is

$$\mathbb{E}[\|Z_t - \tilde{Z}_t\|_1] \leqslant C\exp\left(\int_0^t \mathrm{Lip}(u_s)\mathrm{d}s\right)\ . \qquad (17)$$

Convergence follows, as $C = \mathcal{O}(\|\Delta\|_2^2) = \mathcal{O}(\max_k \Delta_k)$ as $\max_k \Delta_k \to 0$, concluding the proof. □

Example 2. *We can make the discretization error bound explicit when $\pi = \gamma \sim \mathbf{\Gamma}(\alpha, \beta)$. For simplicity, we consider the univariate case. Since $K_s^\gamma(z) = (z+\alpha)/(\beta+s)$, we have $\mathrm{Lip}(u_s) = 1/(\beta+s)$, $u_s(0) = \alpha/(\beta+s)$, $a = 1, b = \alpha$ and $L_k = 1/(\beta+t_k)^2$.*

Furthermore, the inequality (15) *is actually an equality in this specific case, since the denoiser is affine in z. Thus μ_k satisfies the recursion*

$$\begin{aligned}\mu_{k+1} &= \Delta_k\frac{\alpha}{\beta+t_k} + \mu_k\left(1 + \frac{\Delta_k}{\beta+t_k}\right) \\ \mu_{k+1} + \alpha &= (\mu_k+\alpha)\times\frac{\beta+t_{k+1}}{\beta+t_k}\ ,\end{aligned}$$

and as a result, $\mu_k + \alpha = \alpha(\beta+t_k)/\beta$, and finally $\mu_k = t_k\alpha/\beta$. Using (16)*, with $\mathrm{Lip}(u_s) \leqslant 1/(\beta+t_k)$ on $[t_k, t_{k+1}]$, we obtain after simplification*

$$\mathbb{E}[|Z_t - \tilde{Z}_t|] \leqslant \frac{\alpha}{\beta}\sum_{k=0}^{N-1}\frac{\Delta_k^2}{\beta+t_k}\times\left(1+\frac{t}{\beta}\right)\ .$$

We compare this bound with simulations in Figure 3.

Remark 2. *Our sampling procedure based on discretization, Algorithm 1, is quite similar to (Chen & Zhou, 2023, Algorithm 2). Although their construction differs from ours (see Section 2.2), they rigorously derive their algorithm using binomial thinning arguments, while our algorithm leverages the stochastic intensity of counting processes. On the other hand, (Bhattacharya et al., 2025, Algorithm 2) differs substantially from Algorithm 1 and (Chen & Zhou, 2023, Algorithm 2): instead of considering an update of the form $\tilde{Z}_{t_{k+1}} = \tilde{Z}_{t_k} + n_k, n_k \sim \mathrm{Pois}(\Delta_k u_{t_k}(\tilde{Z}_{t_k}))$, with $u_{t_k}(\tilde{Z}_{t_k})$ approximating the Poisson denoiser, they instead draw $\tilde{Z}_{t_{k+1}} \sim \mathrm{Pois}(t_{k+1}u_{t_k}(\tilde{Z}_{t_k}))$. To the best of our understanding, this sampling procedure is not mathematically derived in the paper, and we see no straightforward justification. We assess their sampling procedure with the tractable bivariate Gamma mixture distribution of Appendix E.3, using the same log-uniform time-grid (with $N = 64, 1024$): we compare the quality of approximate π samples obtained with their algorithm and ours, using the analytical expression of the Poisson denoiser. As reported in Table 3, the samples obtained with Algorithm 1 accurately reproduce the first two scalar moments of π, whereas the samples obtained with (Bhattacharya et al., 2025, Algorithm 2) exhibit substantially larger discrepancies.*

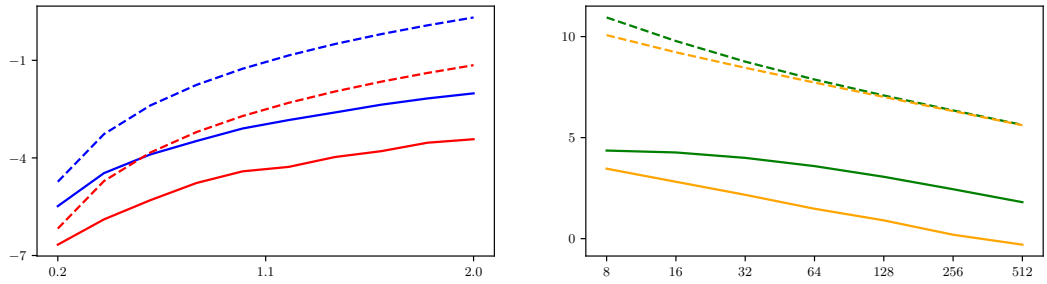

Figure 3: Discretization error for $\mathbf{Z} \sim \mathrm{MPP}(\pi)$, $\pi = \Gamma(\alpha = 3, \beta = 5)$; $\tilde{\mathbf{Z}}$ is the discretized process of Proposition 6. The y-axis is on a log scale. **(Left)** Empirical 1-Wassertein distance (10^7 samples) between $\mathrm{Law}(Z_t)$ and $\mathrm{Law}(\tilde{Z}_t)$ (continuous line) and the associated upper theoretical bound of Example 2 (dashed line) using a uniform time-grid on $[0, t]$, $0.2 \leqslant t \leqslant 2$; with either $N = 16$ (blue) or $N = 64$ (red). We observe that the theoretical upper bound is relatively sharp for low t, with a gap increasing with t. **(Right)** Empirical 1-Wassertein distance between $\mathrm{Law}(Z_t)$ and $\mathrm{Law}(\tilde{Z}_t)$ (continuous line) and the associated theoretical upper bound of Example 2 (dashed line) for $t = 100$, as a function of $N = 8, , 16, \ldots, 512$. We compare the uniform time-grid (green) and a log-uniform grid (orange). More precisely, we use uniformly spaced $\log(\beta + t_k)$: empirically, this yields a substantially smaller error compared to the uniform time-grid.

Sampling method	$N = 64$		$N = 1024$	
	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$
Algorithm 1	5.03	1.95	5.00	1.98
(Bhattacharya et al., 2025, Algorithm 2)	5.41	1.91	5.32	2.12

Table 3: Empirical moments $\hat{\mu}, \hat{\sigma}$ obtained with 10^6 approximate π samples, with π the bivariate Gamma mixture distribution of Appendix E.3. The exact values of $\mu = \mathbb{E}[X_1 + X_2], \sigma^2 = \mathbb{E}[(X_1 + X_2 - \mu)^2]$, are $\mu = 5.0, \sigma \approx 1.99$.

Lemma 1. *Let* $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ *with random rate* X. *Assume* $\mathbb{E}[\|X\|_2^2] < \infty$*; then for every* $t > 0$,

$$\mathscr{W}_2(\mathrm{Law}(Z_t/t), \pi)^2 \leqslant \mathbb{E}[\|Z_t/t - X\|_2^2] = \mathbb{E}[\|X\|_1]/t \ ,$$

where $\mathscr{W}_2$ *is the* 2*-Wasserstein distance defined in Appendix A,* (8).

Proof. Note that for every $t > 0$, conditionally on X, Z_t^i is Poisson distributed with mean and variance tX_i. Assuming $\mathbb{E}[\|X\|_2^2] < \infty$, we have

$$\begin{aligned}
\mathbb{E}\left[\|Z_t - tX\|_2^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\|Z_t - tX\|_2^2 | X\right]\right] \\
&= \sum_{i=1}^{d} \mathbb{E}\left[\mathbb{E}\left[(Z_t^i - tX_i)^2 | X\right]\right] \\
&= t \sum_{i=1}^{d} \mathbb{E}[X_i] = t\mathbb{E}[\|X\|_1] \ .
\end{aligned}$$

Dividing by t^2 gives the result. □

Lemma 2. *Let* $\mathbf{Z}, \mathbf{Y}$ *be counting processes with respective intensities* $t \mapsto u_t(Z_{t-}), t \mapsto v_t(Y_{t-})$*, where* u, v *are measurable, locally integrable functions. Assume that for* $t > 0$*, the integrated discrepancy* $\delta(t) := \int_0^t \mathbb{E}[\|u_s(Y_s) - v_s(Y_s)\|_1]\mathrm{d}s$ *is finite, and finally that* $t \mapsto \mathrm{Lip}(u_t) = \sup_{y \neq z} \|u_t(z) - u_t(y)\|_1 / \|z - y\|_1$ *is locally integrable. Then, for* $t > 0$,

$$\mathscr{W}_{1,1}\left(\mathrm{Law}\left(Z_t\right), \mathrm{Law}\left(Y_t\right)\right) \leqslant \delta(t) \exp\left(\int_0^t \mathrm{Lip}(u_s)\mathrm{d}s\right) \ ,$$

where $\mathscr{W}_{1,1}$ *denotes the* 1*-Wasserstein distance associated to the* l^1 *distance: see Appendix A,* (9).

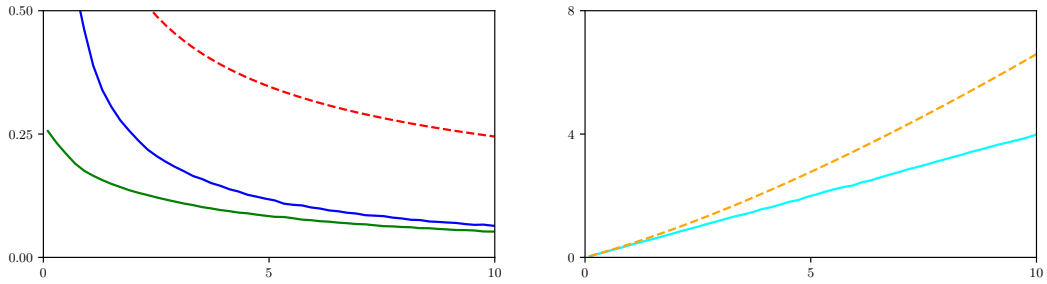

Figure 4: Illustration of Lemma 1 and Lemma 2 with $\mathbf{Z} \sim \mathrm{MPP}(\pi)$, $\pi = \Gamma(\alpha = 3, \beta = 5)$ and $\mathbf{Y} \sim \mathrm{MPP}(\gamma), \gamma = \Gamma(\tilde{\alpha} = 5, \beta = 5)$, for $0 \leqslant t \leqslant 10$. **(Left)** Early stopping error. Blue line shows the empirical 1-Wasserstein distance between $\mathrm{Law}(Z_t/t)$ and π as a function of t, red dashed line is the associated (loose) theoretical bound $\sqrt{\mathbb{E}[X]/t}$, and green line the empirical 1-Wasserstein distance between $K^\pi(Z_t) = (Z_t + \alpha)/(\beta + t)$ and π; see Remark 1. **(Right)** Denoiser estimation error. Cyan line is the empirical 1-Wasserstein distance between $\mathrm{Law}(Z_t)$ and $\mathrm{Law}(Y_t)$, and orange dashed line the theoretical upper bound of Example 3.

Proof. Set $\mathbf{F} = \mathbf{F}^{\mathbf{Z}} \vee \mathbf{F}^{\mathbf{Y}}$. Using the integral representation (10) with $\mathbf{F}$, we may couple $\mathbf{Z}$ and $\mathbf{Y}$ through the same Poisson point process Π. Then, for $1 \leqslant i \leqslant d$,

$$\begin{aligned}
\mathbb{E}[|Z_t^i - Y_t^i|] &\leqslant \mathbb{E}\int_0^t \int_{\mathbb{R}_+} |\mathbb{1}_{[0,u_s^i(Z_s)]}(x) - \mathbb{1}_{[0,v_s^i(Y_s)]}(x)| \Pi(\mathrm{d}s, \mathrm{d}x) \\
&= \mathbb{E}\int_0^t \int_{\mathbb{R}_+} |\mathbb{1}_{[0,u_s^i(Z_s)]}(x) - \mathbb{1}_{[0,v_s^i(Y_s)]}(x)| \mathrm{d}x\mathrm{d}s \\
&= \mathbb{E}\int_0^t |u_s^i(Z_s) - v_s^i(Y_s)| \mathrm{d}s \\
&\leqslant \int_0^t \mathbb{E}[|u_s^i(Y_s) - v_s^i(Y_s)|] \mathrm{d}s + \int_0^t \mathbb{E}[|u_s^i(Z_s) - u_s^i(Y_s)|] \mathrm{d}s \ ,
\end{aligned}$$

where the second line uses the compensation formula of Π, see, e.g., (Kyprianou, 2014, Theorem 4.4). Summing for $1 \leqslant i \leqslant d$ yields

$$\begin{aligned}
\mathbb{E}[\|Z_t - Y_t\|_1] &= \sum_{i=1}^d \mathbb{E}[|Z_t^i - Y_t^i|] \\
&\leqslant \int_0^t \mathbb{E}[\|u_s(Y_s) - v_s(Y_s)\|_1] \mathrm{d}s + \int_0^t \mathbb{E}[\|u_s(Z_s) - u_s(Y_s)\|_1] \mathrm{d}s \\
&\leqslant \delta(t) + \int_0^t \mathbb{E}[\|Z_s - Y_s\|_1] \,\mathrm{Lip}(u_s) \mathrm{d}s \ .
\end{aligned}$$

Gronwall's inequality leads to the upper bound on $\mathbb{E}[\|Z_t - Y_t\|_1]$, and thus on the Wasserstein distance. □

Example 3. *In dimension 1, if* $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ *with* $\pi = \gamma = \mathbf{\Gamma}(\alpha, \beta)$*, the denoiser is an affine function of* z*:* $K_t^\gamma(z) = \frac{z+\alpha}{t+\beta}$*; see Appendix B.4. Let* $\mathbf{Y} \sim \mathrm{MPP}(\tilde{\gamma})$*, with* $\tilde{\gamma} \sim \mathbf{\Gamma}(\tilde{\alpha}, \beta)$*; then* $K_t^{\tilde{\gamma}}(z) = \frac{y+\tilde{\alpha}}{t+\beta}$*. We have* $\delta(t) = \int_0^t \mathbb{E}[|K_s^\gamma(Y_s) - K_s^{\tilde{\gamma}}(Y_s)|] \mathrm{d}s = |\alpha - \tilde{\alpha}| \log(1 + t/\beta)$ *and* $\mathrm{Lip}(K_t^\gamma) = 1/(t+\beta)$*, hence the bound* $\mathscr{W}_{1,1}(\mathrm{Law}(Z_t), \mathrm{Law}(Y_t)) \leqslant |\alpha - \tilde{\alpha}| \left(1 + \frac{t}{\beta}\right) \log\left(1 + \frac{t}{\beta}\right)$*. Here, the exact value of* $\mathscr{W}_{1,1}(\mathrm{Law}(Z_t), \mathrm{Law}(Y_t))$ *is tractable, and reads* $\frac{|\alpha-\tilde{\alpha}|}{\beta} t$*. The general bound we computed differs from the actual distance by only a logarithmic factor; see also Figure 4.*

Combining the previous results yields the upper bounds below. Using thinning (Algorithm 2) introduces no additional error, leading to (18); however, execution time is uncertain. Using time

discretization instead (Algorithm 1) does not have this drawback, but introduces the additional error derived in Proposition 6, yielding (19).

Corollary 1. *Let $\mathbf{Z} \sim \mathrm{MPP}(\pi)$ with random rate X, and consider an approximation $\mathbf{Y}$ with stochastic intensity $k^\theta \approx K^\pi$. Assume $\mathbb{E}[\|X\|_2^2] < \infty$, and that k^θ satisfies the assumptions of Lemma 2. Then, for any fixed $t > 0$, using the notation of Lemma 2,*

$$\mathscr{W}_{1,1}(\mathrm{Law}(Y_t/t), \pi) \leqslant \sqrt{d\frac{\mathbb{E}[\|X\|_1]}{t}} + \frac{\delta(t)}{t} \exp\left(\int_0^t \mathrm{Lip}(k_s^\theta)\mathrm{d}s\right) . \tag{18}$$

If $\tilde{\mathbf{Y}}$ is the counting process obtained by discretization of $\mathbf{Y}$ on a time-grid $0 = t_0 < t_1 < \cdots < t_N$, and if k^θ satisfies the regularity assumptions of Proposition 6, then

$$\mathscr{W}_{1,1}(\mathrm{Law}(\tilde{Y}_t/t), \pi) \leqslant \sqrt{d\frac{\mathbb{E}[\|X\|_1]}{t}} + \frac{1}{t}\left(\delta(t) + C\right) \exp\left(\int_0^t \mathrm{Lip}(k_s^\theta)\mathrm{d}s\right) , \tag{19}$$

with $C = \mathcal{O}(\max_k \Delta_k)$ being defined in the proof of Proposition 6.

Proof. Using the triangle inequality,

$$\begin{aligned}\mathscr{W}_{1,1}(\mathrm{Law}(Y_t/t), \pi) &\leqslant \mathbb{E}[\|Y_t/t - X\|_1] \\ &\leqslant \mathbb{E}[\|Z_t/t - X\|_1] + \mathbb{E}[\|Z_t - Y_t\|_1]/t \, .\end{aligned}$$

The first term is upper bounded with Lemma 1:

$$\begin{aligned}\mathbb{E}[\|Z_t/t - X\|_1] &\leqslant \sqrt{d}\mathbb{E}[\|Z_t/t - X\|_2] \\ &\leqslant \sqrt{d}\sqrt{\frac{\mathbb{E}[\|X\|_1]}{t}}\end{aligned}$$

Plugging in the bounds of Lemma 1 and Lemma 2 yields (18). Then,

$$\begin{aligned}\mathscr{W}_{1,1}(\mathrm{Law}(\tilde{Y}_t/t), \pi) &\leqslant \mathbb{E}[\|\tilde{Y}_t/t - X\|_1] \\ &\leqslant \mathbb{E}[\|Z_t/t - X\|_1] + \mathbb{E}[\|Z_t - \tilde{Y}_t\|_1]/t \\ &\leqslant \mathbb{E}[\|Z_t/t - X\|_1] + \mathbb{E}[\|Z_t - Y_t\|_1]/t + \mathbb{E}[\|\tilde{Y}_t - Y_t\|_1]/t\end{aligned}$$

yields (19) using Proposition 6, Equation (17). □

B.4 The Gamma case

Diffusion-based methods usually consider Gaussian mixtures as toy examples. For MPDM, the tractable case corresponds to Gamma mixtures.

MPP with Gamma random rate. Consider a mixed Poisson process $\mathbf{Z}$ with distribution $\mathrm{MPP}(\gamma)$, where

$$\gamma(\mathrm{d}x) = \mathbf{\Gamma}(\alpha, \beta) = \prod_{i=1}^d \mathbf{\Gamma}(x_i; \alpha_i, \beta_i)\mathrm{d}x_i = \prod_{i=1}^d \mathrm{e}^{-\beta_i x_i} x_i^{\alpha_i - 1} \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)}\mathrm{d}x_i \, ,$$

with $\alpha, \beta \in (0, +\infty)^d$. To be clear, $\mathbf{\Gamma}(\alpha, \beta)$ is the multivariate Gamma distribution, a probability measure on $\mathbb{R}_+^d$, $x_i \in \mathbb{R}_+ \mapsto \mathbf{\Gamma}(x_i; \alpha_i, \beta_i)$ is the density of the univariate Gamma distribution, and Γ denotes the Gamma function.

Then each component $\mathbf{Z}_i, 1 \leqslant i \leqslant d$ is a *Póyla process*, see (Grandell, 1997, Example 4.1). Specifically, for every $t \geqslant 0$, the components of Z_t are independent and admit a negative binomial distri-

bution:

$$p_t^\gamma(z) \coloneqq \mathbb{P}(Z_t = z) = \int_{\mathbb{R}_+} \prod_{i=1}^d \mathrm{e}^{-tx_i} \frac{(tx_i)^{z_i}}{z_i!} \gamma(\mathrm{d}x) \tag{20}$$
$$= \prod_{i=1}^d \int_0^\infty \mathrm{e}^{-(t+\beta_i)x_i} \frac{x_i^{z_i+\alpha_i-1}}{z_i!\Gamma(\alpha_i)} t^{z_i} \beta_i^{\alpha_i} \mathrm{d}x_i$$
$$= \prod_{i=1}^d \frac{\Gamma(z_i+\alpha_i)}{(t+\beta_i)^{z_i+\alpha_i}} \frac{t^{z_i}\beta_i^{\alpha_i}}{z_i!\Gamma(\alpha_i)}$$
$$= \prod_{i=1}^d \frac{\Gamma(z_i+\alpha_i)}{z_i!\Gamma(\alpha_i)} \left(1 - \frac{\beta_i}{t+\beta_i}\right)^{z_i} \left(\frac{\beta_i}{t+\beta_i}\right)^{\alpha_i} ,$$

where we recognize unnormalized Gamma densities to compute the integrals. Given $z \in \mathbb{N}^d$, the denoising density (*i.e.*, the density of X conditionally on $Z_t = z$) is a product of Gamma densities:

$$\pi_t(\mathrm{d}x|z) = p_t^\gamma(x|z)\mathrm{d}x \coloneqq \prod_{i=1}^d \mathrm{e}^{-tx_i} \frac{(tx_i)^{z_i}}{z_i!} \times \frac{\gamma(\mathrm{d}x)}{\mathbb{P}(Z_t = z)} \tag{21}$$
$$\propto \mathbf{\Gamma}(z+\alpha, t\mathbf{1}+\beta) .$$

The denoiser is the mean of this distribution, hence:

$$[K_t^\gamma(z)]_i = \frac{z_i+\alpha_i}{t+\beta_i},\ 1 \leqslant i \leqslant d .$$

Mixture of Gamma. If $\mathbf{Z} \sim \mathrm{MPP}(\mu)$ with $\mu(\mathrm{d}x) = \sum_{j=1}^N w_j \prod_{i=1}^d \mathbf{\Gamma}(x_i; \alpha_{ij}, \beta_{ij})\mathrm{d}x_i$ with w on the simplex, (20) is directly adapted by adding the weighted sum to the equation:

$$p_t^\mu(z) = \sum_{j=1}^N w_j \prod_{i=1}^d \frac{\Gamma(z_i+\alpha_{ij})}{z_i!\Gamma(\alpha_{ij})} \left(1 - \frac{\beta_{ij}}{t+\beta_{ij}}\right)^{z_i} \left(\frac{\beta_{ij}}{t+\beta_{ij}}\right)^{\alpha_{ij}} .$$

The denoising density is a Gamma mixture, namely:

$$p_t^\mu(x|z)\mathrm{d}x = \sum_{j=1}^N \tilde{w}_j \prod_{i=1}^d \mathbf{\Gamma}(x_i; z_i+\alpha_{ij}, t+\beta_{ij})\mathrm{d}x_i \ , \tilde{w}_j \coloneqq \frac{w_j}{p_t^\mu(z)} \prod_{i=1}^d \frac{t^{z_i}\beta_{ij}^{\alpha_{ij}}}{\Gamma(\alpha_{ij})z_i!} \frac{\Gamma(z_i+\alpha_{ij})}{(t+\beta_{ij})^{z_i+\alpha_{ij}}} ,$$

hence the denoiser

$$[K_t^\mu(z)]_i = \sum_{j=1}^N \tilde{w}_j \frac{z_i+\alpha_{ij}}{t+\beta_{ij}} \ , 1 \leqslant i \leqslant d .$$

C Overwiew of diffusion-based posterior sampling methods

We provide a brief overview of diffusion-based posterior sampling. In particular, we explain the core ideas of the algorithms that are adapted to the MPDM setting in Section 3. We will refer to the diffusion model introduced in Section 2.2 and the associated notations.

Gaussian inverse problems. Let $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^{d_y}$ denote the forward operator, with $1 \leqslant d_y \leqslant d$, and consider the posterior distribution π^y of $X \sim \pi$ given the observation $Y = y$, where

$$Y = \mathcal{A}(X) + \sigma\varepsilon ,$$

with $\varepsilon \sim \mathrm{N}(0, \mathrm{I}_d)$ and $\sigma > 0$. From Bayes' rule, we have $\pi^y(\mathrm{d}x) \propto g(y|x)\pi(\mathrm{d}x)$, with g the likelihood function: in this case, $g(y|x) = \mathrm{N}(y; \mathcal{A}(x), \sigma^2\mathrm{I}_d)$.

Instead of focusing on a specific forward operator $\mathcal{A}$, recent training-free methods use a pre-trained diffusion model as a general-purpose prior. The Gaussian denoiser D^π and the observation y are combined to sample from the posterior distribution π^y of a given inverse problem; see the review by Janati et al. (2025b).

Guidance-based methods. Granted access to the *conditional denoiser*, *i.e.*, D^{π^y}, samples from π^y could be obtained by simulating the associated diffusion (5). Diffusion Posterior Sampling (DPS) (Chung et al., 2023), approximates the conditional denoiser D^{π^y}[1] as

$$D_t^{\pi^y}(x_t) = D_t^{\pi}(x_t) + \nabla \log g_t(y|x_t) \approx D_t^{\pi}(x_t) + \nabla \log g(y|D_t^{\pi}(x_t)) \,, \tag{22}$$

where the intractable term $g_t(y|x_t) \coloneqq \int g(y|x)\pi_t(\mathrm{d}x|x_t)$ is estimated by approximating $\pi_t(\mathrm{d}x|x_t) \approx \delta_{D_t^{\pi}(x_t)}(\mathrm{d}x)$. In practice, evaluating this approximation requires differentiating the learned denoiser with respect to x_t. Alternatively, ΠGDM (Song et al., 2023) assumes $\pi_t(\mathrm{d}x|x_t)$ is Gaussian distributed with mean $D_t^{\pi}(x_t)$ and covariance matrix $r_t \mathrm{I}_d$, with $r_t > 0$. For Gaussian inverse problems with linear $\mathcal{A}$, this approximation makes the conditional score, or equivalently the conditional denoiser D^{π^y}, tractable. However, ΠGDM does not transpose to Poisson inverse problems.

Split Gibbs sampling methods. Instead of approximating D^{π^y} to simulate a diffusion process directly, another class of methods sample from a sequence of intermediary, augmented distributions. For instance, PnP-DM (Wu et al., 2024) consider the augmented distributions

$$\pi_k^y(\mathrm{d}x, \mathrm{d}\tilde{x}) \propto g(y|\tilde{x})\mathrm{e}^{-\|x-\tilde{x}\|_2^2/(2\rho_k^2)}\pi(\mathrm{d}x) \,, \tag{23}$$

where $(\rho_k)_{k=1}^n \in (0, +\infty)^n$ is typically non-increasing. For each $1 \leqslant k \leqslant n$, $x, \tilde{x}$ are updated through a two steps procedure. The *likelihood step*, *i.e.*, sampling from $\pi_k^y(\mathrm{d}\tilde{x}|x) \propto g(y|\tilde{x})\mathrm{e}^{-\|x-\tilde{x}\|_2^2/(2\rho_k^2)}$, is exact for Gaussian inverse problem with linear $\mathcal{A}$; in general, it can be tackled with standard MCMC methods. The *prior step* leverages the pre-trained diffusion model, as

$$\pi_k^y(\mathrm{d}x|\tilde{x}) \propto \mathrm{e}^{-\|x-\tilde{x}\|_2^2/(2\rho_k^2)}\pi(\mathrm{d}x) \propto \pi_{t_k}\left(\mathrm{d}x\,|t_k\tilde{x}\right) \,,$$

setting $t_k \coloneqq 1/\rho_k^2$: the diffusion process is simulated from t_k onward, with initial state $t_k\tilde{x}$. As ρ_k approaches zero, $x \approx \tilde{x}$ is approximately distributed as π^y.

A related approach is DAPS (Zhang et al., 2025): its structure resembles that of PnP-DM, but includes a third, *noising* step that decouples successive samples to promote a larger exploration of the posterior distribution. We describe DAPS using the diffusion model of Section 2.2, which slightly modifies the formulation of the original paper (which relied on a forward/backward diffusion formalism). This algorithm revolves around the sequence of distributions

$$\pi_{t_k}^y(\mathrm{d}x|x_{t_k}) \propto g(y|x)\pi_{t_k}(\mathrm{d}x|x_{t_k}) \,, \tag{24}$$

on a time-grid $t_1 < \ldots t_N$. DAPS uses the approximation $\pi_t(\mathrm{d}x|x_t) \approx \mathrm{N}(\hat{x}(x_t), r_t^2\mathrm{I}_d)$, see (Zhang et al., 2025, Equation 5). Here, $\hat{x}(x_t)$ denotes an estimator of X given $X_t = x_t$: it can be obtained as $D_T^{\pi}(x_T)$, where x_T is computed by numerical integration of (5) from t up to a terminal time T using a given time-grid. Using only one denoising step results in the simple estimator $\hat{x}(x_t) = D_t^{\pi}(x_t)$; this design choice is discussed in (Zhang et al., 2025, Appendix I). By analogy with PnP-DM, we refer to the estimation of $\hat{x}(x_t)$, for a given $t = t_k$, as the *prior step*. Then, the *likelihood step* consists of sampling from

$$\pi_{t_k}^y(\mathrm{d}x|x_{t_k}) \propto g(y|x)\mathrm{N}(\hat{x}(x_{t_k}), r_{t_k}^2\,\mathrm{I}_d)$$

with an MCMC scheme. Finally, given the output x of the likelihood step, one draws $x_{t_{k+1}} = t_{k+1}x + \sqrt{t_{k+1}}G_{k+1}$, with $G_{k+1} \sim \mathrm{N}(0, \mathrm{I}_d)$. This *noising step* is meant to decouple the successive x_{t_k}: instead of generating a single trajectory of the denoising process $\mathbf{X}^{\mathrm{d}}$ (5), we sample a "fresh" noisy intermediate $x_{t_{k+1}}$ through the noising process $\mathbf{X}^{\mathrm{n}}$.

Gaussian denoising. Diffusion-based posterior sampling methods often exploit the Gaussian nature of the inverse problem, leveraging the matching noise structure between the diffusion and the inverse problem; e.g., (Song et al., 2023; Kawar et al., 2022). This correspondence is best illustrated when the problem consists of pure Gaussian noise, *i.e.*, $\mathcal{A} = a\mathrm{I}_d$ with $a \in \mathbb{R}$: then,

$$D_t^{\pi^y}(x_t) = D_{t+(a/\sigma)^2}^{\pi}\left(x_t + ay/\sigma^2\right) \,. \tag{25}$$

In other words, sampling from the posterior distribution π^y is possible by simulating (5) while shifting the inputs of D^{π}, without any further approximation.

[1] This approximation is usually formulated in terms of the diffusion score instead of the denoiser: both formulations are equivalent thanks to Tweedie's formula.

Poisson inverse problems. If $X \sim \pi$ with π supported on $\mathbb{R}_+^d$ and $\mathcal{A} : \mathbb{R}_+^d \to \mathbb{R}_+^{d_y}$, a Poisson inverse problem consists of sampling from the posterior distribution $\pi^y \propto g(y|x)\pi(\mathrm{d}x)$ of X given $Y = y$, where $Y \sim \mathrm{Pois}(\mathcal{A}(X))$, *i.e.*,

$$g(y|x) \propto \prod_{i=1}^{d_y} \mathrm{e}^{-[\mathcal{A}(x)]_i} [\mathcal{A}(x)]_i^{y_i} \ . \tag{26}$$

This class of problems is mostly excluded from the aforementioned posterior sampling methods. Even DPS, whose approximation (22) theoretically holds for any likelihood g, eventually replaces the Poisson likelihood by a "Gaussianized" one due to poor numerical stability; see (Chung et al., 2023, Appendix C.4). This approximation is only valid when the measured values y_i are large, excluding the regime where some y_i are close to or equal to zero. Gibbs-sampling methods can theoretically handle the Poisson likelihood, but the likelihood step (23) requires a non-trivial MCMC sampling of a density that is supported on $\mathbb{R}_+^d$. Moreover, standard gradient-based methods such as Langevin MCMC are problematic near the boundary of the positive orthant, since the Poisson log-likelihood has a singular gradient when $[\mathcal{A}(x)]_i \to 0$ for $y_i > 0$; and unconstrained Gaussian updates can leave the domain.

In stark contrast with the Gaussian case, even when $\mathcal{A} = a\mathrm{I}_d$, there is no straightforward diffusion-based posterior sampling method. Leveraging Theorem 3, PDPS precisely fills this gap.

D Algorithmics details

D.1 Training

Algorithm 3 Training

Require: Dataset $\mathcal{D}$, trainable network k^θ, strictly convex function $\phi : \mathbb{R}_+^d \to \mathbb{R}$, mini-batch size B, number of epoch N, probability measure $\nu(\mathrm{d}t)$ on $\mathbb{R}_+$ and weight function w.

1: Initialize θ
2: **repeat**
3: Draw mini-batch $x_1, \dots, x_B \in \mathcal{D}$
4: Sample $t_1, \dots, t_B \sim \nu(\mathrm{d}t)$
5: Draw $z_i \sim \mathrm{Pois}(t_i x_i), 1 \leqslant i \leqslant B$ ▷ Noising
6: Compute $k_{t_i}^\theta(z_i), 1 \leqslant i \leqslant B$ ▷ Denoising
7: Perform gradient descent step on $\sum_{i=1}^B \mathscr{D}_\phi(x_i, k_{t_i}^\theta(z_i))w(t_i)$ ▷ Minimization of $\mathscr{L}(\theta)$
8: **until** convergence or target number of epochs is reached
9: **return** learned model k^θ

For completeness, we provide the training pseudo-code that outputs an approximation of the Poisson denoiser, $k^\theta \approx K^\tau$. This algorithm is similar to the one proposed by Bhattacharya et al. (2025); Chen & Zhou (2023), albeit with a generic Bregman divergence instead of (4). For our numerical experiments, we observed that using the usual Euclidean distance instead of (4) had little influence on the learned model performance. The distribution $\nu(\mathrm{d}t)$ can be either discrete, if one fixes a time-grid, or continuous, e.g., uniform on some $[t_{\min}, t_{\max}]$. In practice, we set ν as the log-uniform distribution on $[t_{\min}, t_{\max}]$ and the weight function $t \mapsto w(t) = t$, similar to Bhattacharya et al. (2025). This increases the number of low-SNR (*i.e.*, low t) inputs z_t while preserving the training objective associated with uniform t:

$$\mathscr{L}(\theta) = \int_{t_{\min}}^{t_{\max}} \mathbb{E}[\mathscr{D}_\phi(X, k_t^\theta(Z_t))]\mathrm{d}t \propto \int_{t_{\min}}^{t_{\max}} w(t)\mathbb{E}[\mathscr{D}_\phi(X, k_t^\theta(Z_t))]\nu(\mathrm{d}t) \ .$$

We detail the exact parameters used in our numerical experiments in Appendix E.

D.2 Poisson denoising with PDPS

PDPS (Algorithm 4) is a relatively straightforward implementation of Theorem 3 that solves the inverse problem defined by $Y \sim \mathrm{Pois}(aX), X \sim \pi$ by leveraging a pre-trained MPDM. We provide the proof and pseudo-code below, and show a sample trajectory in Figure 5.

Algorithm 4 Poisson denoising posterior sampling (PDPS)

Require: Learned denoiser $k^\theta \approx K^\pi$, rate $a > 0$, observation $y \in \mathbb{N}^d$, time-grid $0 = t_0 < t_1 < \cdots < t_N$, with $\Delta_k = t_{k+1} - t_k$

1: Initialize $z_0 = 0$
2: **for** $k = 0, \dots, N-1$ **do**
3: Estimate $K_{t_k}^{\pi^y}(z_k) \approx \tilde{K}_{t_k}^{\pi^y}(z_k) = k_{t_k+a}^\theta(z_k + y)$ ▷ Theorem 3
4: Draw $n_k \sim \mathrm{Pois}(\Delta_k \tilde{K}_{t_k}^{\pi^y}(z_k))$
5: Set $z_{k+1} = z_k + n_k$ ▷ Discretization scheme (Algorithm 1)
6: **end for**
7: **return** z_N/t_N or $k_{t_{N-1}+a}^\theta(z_{N-1} + y)$ ▷ See Remark 1

Theorem 3. *Let π^y denote the posterior distribution of X given $Y = y$, where $Y \sim \mathrm{Pois}(aX)$ and $X \sim \pi$. The denoiser K^{π^y} associated with $\mathrm{MPP}(\pi^y)$ is given by $(t, z) \mapsto K_t^{\pi^y}(z) = K_{t+a}^\pi(z+y)$.*

Proof. Fix $t > 0$ and $z \in \mathbb{R}_+^d$. Recall that, from Proposition 2, we have

$$K_t^\pi(z) = \int_{\mathbb{R}_+^d} x \prod_{j=1}^d x_j^{z_j} \mathrm{e}^{-tx_j} \pi(\mathrm{d}x) \Big/ \int_{\mathbb{R}_+^d} \prod_{j=1}^d x_j^{z_j} \mathrm{e}^{-tx_j} \pi(\mathrm{d}x) \, .$$

Similarly, the denoiser associated to $\mathrm{MPP}(\pi^y)$ writes

$$\begin{aligned} K_t^{\pi^y}(z) &= \int_{\mathbb{R}_+^d} x \prod_{j=1}^d x_j^{z_j} \mathrm{e}^{-tx_j} \pi^y(\mathrm{d}x) \Big/ \int_{\mathbb{R}_+^d} \prod_{j=1}^d x_j^{z_j} \mathrm{e}^{-tx_j} \pi^y(\mathrm{d}x) \\ &= \int_{\mathbb{R}_+^d} x \prod_{j=1}^d x_j^{z_j} \mathrm{e}^{-tx_j} \prod_{k=1}^d \mathrm{e}^{-ax_k} (ax_k)^{y_k} \pi(\mathrm{d}x) \Big/ \int_{\mathbb{R}_+^d} \prod_{j=1}^d x_j^{z_j} \mathrm{e}^{-tx_j} \prod_{k=1}^d \mathrm{e}^{-ax_k} (ax_k)^{y_k} \pi(\mathrm{d}x) \\ &= \int_{\mathbb{R}_+^d} x \prod_{j=1}^d x_j^{z_j+y_j} \mathrm{e}^{-(t+a)x_j} \pi(\mathrm{d}x) \Big/ \int_{\mathbb{R}_+^d} \prod_{j=1}^d x_j^{z_j+y_j} \mathrm{e}^{-(t+a)x_j} \pi(\mathrm{d}x) \, , \end{aligned}$$

using $\pi^y(\mathrm{d}x) \propto g(y|x)\pi(\mathrm{d}x) \propto \prod_{k=1}^d \mathrm{e}^{-ax_k}(ax_k)^{y_k}\pi(\mathrm{d}x)$. We recognize the expression of $K_{t+a}^\pi(z+y)$, concluding the proof. □

D.3 MPDM-BASED DPS

We provide additional details regarding the MPDM adaptation of DPS.

Theorem 4. *Consider the posterior distribution π^y of a general Poisson inverse problem with likelihood function g defined in* (26)*. For any $z_t \in \mathbb{N}^d$ and $t > 0$, writing $e_i = (\delta_{i,j})_{1 \leqslant j \leqslant d}$, we have*

$$[K_t^{\pi^y}(z_t)]_i = [K_t^\pi(z_t)]_i \frac{g_t(y|z_t + e_i)}{g_t(y|z_t)} = [K_t^\pi(z_t)]_i \left(1 + \frac{g_t(y|z_t + e_i) - g_t(y|z_t)}{g_t(y|z_t)}\right) \, , 1 \leqslant i \leqslant d \, .$$

Proof. Let $\mathbf{Z} \sim \mathrm{MPP}(\pi)$. As seen in the proof of Proposition 2, we have for every $z_t \in \mathbb{N}^d, t > 0$,

$$[K_t^\pi(z_t)]_i = \frac{z_t^i + 1}{t} \frac{\mathbb{P}(Z_t = z_t + e_i)}{\mathbb{P}(Z_t = z_t)} \, , \tag{27}$$

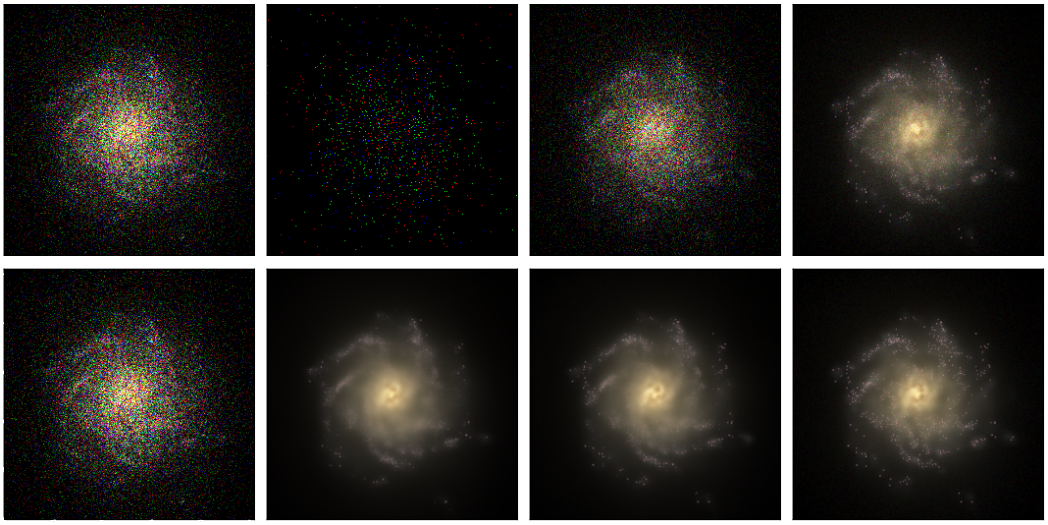

Figure 5: Intermediate states of PDPS for the Poisson denoising problem on the EAGLE dataset. **(Top-left, bottom-left)** Observation $y \sim \mathrm{Pois}(ax)$, with x an image of the validation dataset. **(Top row)** $Z_t^y/(1+t)$, with $\mathbf{Z}^y \sim \mathrm{MPP}(\pi^y)$, for increasing values of t. This MPP is simulated using the Poisson denoiser approximation $K^{\pi^y}(Z_t^y) \approx k^\theta(Z_t^y + y)$. **(Bottom-row)** The corresponding Poisson denoiser outputs $K^{\pi^y}(Z_t^y) \approx k^\theta(Z_t^y + y)$.

Algorithm 5 MPDM adaptation of DPS

Require: Learned denoiser $k^\theta \approx K^\pi$, likelihood $g(y|\cdot)$, measurement y, time-schedule $0 = t_0 < t_1 < \cdots < t_N$ with $\Delta_k = t_{k+1} - t_k$.

1: Initialize $z_0 = 0$
2: **for** $k = 0, \ldots, N-1$ **do**
3: Compute $k_{t_k}^\theta(z_k)$
4: Estimate $K_{t_k}^{\pi^y}(z_k) \approx \tilde{K}_{t_k}^{\pi^y}(z_k)$ using $g(y|\cdot)$ and (6) (either the continuous approximation or the independent coordinates approximation)
5: Draw $n_k \sim \mathrm{Pois}(\Delta_k \tilde{K}_{t_k}^{\pi^y}(z_k))$
6: Set $z_{k+1} = z_k + n_k$ ▷ Discretization scheme (Algorithm 1)
7: **end for**
8: **return** z_N/t_N or $\tilde{K}_{t_{N-1}}^{\pi^y}(z_{N-1})$ ▷ See Remark 1

where z_t^i denotes the i-th coordinate of z_t. If $\mathbf{Z}^y \sim \mathrm{MPP}(\pi^y)$, we have

$$\begin{aligned}
\mathbb{P}(Z_t^y = z_t) &= \int_{\mathbb{R}_+^d} \prod_{j=1}^d \mathrm{e}^{-t x_j} \frac{(t x_j)^{z_t^j}}{z_t^j!} \pi^y(\mathrm{d}x) \\
&\propto \int_{\mathbb{R}_+^d} \prod_{j=1}^d \mathrm{e}^{-t x_j} \frac{(t x_j)^{z_t^j}}{z_t^j!} g(y|x) \pi(\mathrm{d}x) \\
&\propto \mathbb{P}(Z_t = z) \int_{\mathbb{R}_+^d} g(y|x) \pi_t(\mathrm{d}x|z_t) \,,
\end{aligned}$$

using $\pi^y(\mathrm{d}) \propto g(y|x)\pi(\mathrm{d}x)$ and $p_t(z_t)\pi_t(\mathrm{d}x|z_t) = p_t(z_t|x)\pi(\mathrm{d}x)$. Combining (27) and $g_t(y|z_t) = \int_{\mathbb{R}_+^d} g(y|x)\pi_t(\mathrm{d}x|z_t)$, this yields

$$[K_t^{\pi^y}(z_t)]_i = [K_t^\pi(z_t)]_i \frac{g_t(y|z_t + e_i)}{g_t(y|z_t)} = [K_t^\pi(z_t)]_i \left(1 + \frac{g_t(y|z_t + e_i) - g_t(y|z_t)}{g_t(y|z_t)}\right) .$$

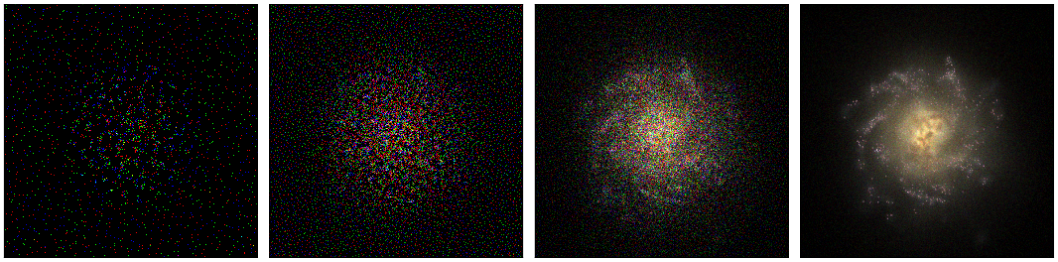

Figure 6: Trajectory generated by MDPM-based DPS, for the Poisson denoising problem. Images show $Z_t^y/(1+t)$, with $\mathbf{Z}^y \sim \mathrm{MPP}(\pi^y)$, for increasing values of t. This MPP is simulated using the Poisson denoiser approximation of Theorem 4, with the discrete derivative being approximated by the continuous one.

□

Estimation of the discrete derivative. Computing the discrete derivative resulting from the DPS approximation, that is, equation (6), would require d denoiser evaluations, which is prohibitive in most applications. Thus, we suggest two approximations.

The first replaces the discrete derivative with a continuous one, using Proposition 2 to extend the domain of definition of K_t^π to $\mathbb{R}_+^d$. We motivate this approximation by noting that, by construction of the MPP, for a relatively large $t > 0$, K_t^π is typically evaluated with an input $z_t \approx tx$, with $x = \mathcal{O}(1)$. Then, $K_t^\pi(z_t + e_i) \approx K_t^\pi(t(x + e_i/t)) \approx K^\pi(tx) + \partial_i K_t^\pi(tx)$, hence $K_t^\pi(z_t + e_i) - K_t^\pi(z_t) \approx \partial_i K_t^\pi(z_t)$.

If for any $1 \leqslant i \leqslant d$, $z_t \mapsto [K_t^\pi(z_t)]_i$ only depends on z_t^i, we may evaluate the discrete derivative with only two denoiser evaluations: indeed, $[K_t^\pi(z_t + e_j)]_i - [K_t^\pi(z_t)]_i = ([K_t^\pi(z_t + 1)]_i - [K_t^\pi(z_t)]_i) \times \delta_{i,j}$ in that case. This condition holds notably if $\pi = \pi_1 \otimes \cdots \otimes \pi_d$. However, this assumption is generally too restrictive in practice and therefore introduces an approximation error.

As noted in Example 1, both approximations are in fact exact when $\pi = \gamma \sim \mathbf{\Gamma}(\alpha, \beta)$, as $[K_t^\gamma(z_t)]_i = (z_t^i + \alpha_i)/(\beta_i + t)$. Since the continuous approximation can be justified for large t, a reasonable strategy would be to use it for large t, and default to the independent coordinates approximation when t is close to 0. We illustrate this compromise in Figure 7, where we compare the quality of both approximations on tractable examples. For the numerical experiments of Section 4, we observed that the continuous approximation generally led to better empirical results, but that starting with the independent coordinates approximation for small t usually slightly improved results; see Appendix E.3. These observations suggest that the continuous derivative should be used by default, while the independent coordinate approximation can be relevant for small t, depending on the application.

D.4 MPDM-BASED PNP-DM

Consider the following family of distributions. For clarity, we keep the notation ρ_k of the original PnP-DM paper; however, in practice we set $t_k = 1/\rho_k^2$, for a time-grid $0 \approx t_0 < t_1 < \cdots < t_N$:

$$\pi_k^y(\mathrm{d}x, \mathrm{d}\tilde{x}) \propto g(y|\tilde{x}) \exp\left(-\mathscr{D}_{\mathrm{e}}(\tilde{x}, x)/\rho_k^2\right) \pi(\mathrm{d}x)\mathrm{d}\tilde{x} \ .$$

The marginal in x is heuristically expected to approach the posterior π^y as $t_k \to \infty$. We do not claim any exact convergence: this expression is motivated by the analogy existing with the Gaussian diffusion case, as Proposition 3 shows that the denoising distribution of $\mathrm{MPP}(\pi)$ naturally appears within the prior step. However, a properly normalized kernel of this form generally involves an additional normalization factor, potentially depending on x, which is omitted here.

Prior step. Proposition 3 shows that $\pi_k^y(\mathrm{d}x|\tilde{x}) \propto \pi_{t_k}(\mathrm{d}x\,|t_k\tilde{x})$, setting $t_k := 1/\rho_k^2$. Recall that $\pi_t(\mathrm{d}x|z_t)$ denotes the conditional distribution of X given $Z_t = z_t$, with $\mathbf{Z}$ the MPP with rate X. A priori, it is defined for $z_t \in \mathbb{N}^d$; however, equation (2) allows us to extend its domain of

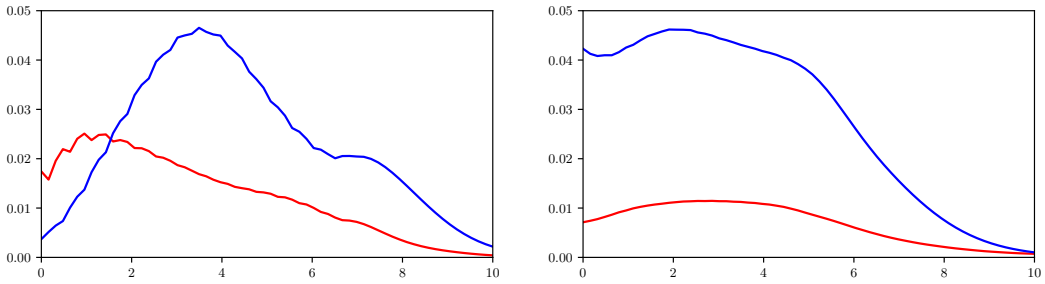

Figure 7: Errors associated with the approximation of the Poisson denoiser K_t^π discrete derivative, for $0 \leqslant t \leqslant 10$, when π is a mixture of bivariate Gamma distributions. Red lines show the total error associated with the continuous approximation, *i.e.*, $1/4 \sum_{i,j=1}^2 |[K_t^\pi(z_t + e_j)]_i - [K_t^\pi(z_t)]_i - \partial_j [K_t^\pi(z_t)]_i|$. Blue lines show the error associated with the independent coordinate approximations, that is, $1/4 \sum_{i,j=1}^2 |[K_t^\pi(z_t + e_j)]_i - [K_t^\pi(z_t)]_i - ([K_t^\pi(z_t + \mathbf{1})]_i - [K_t^\pi(z_t)]_i) \times \delta_{ij}|$. The errors are averaged over $z_t = (k, l), 1 \leqslant k, l \leqslant 15$. **(Left)** π is the distribution used in the toy example; see Appendix E.3. We observe that the independent coordinates approximation is preferable in the low t regime, whereas the continuous approximation gets significantly better as t increases. **(Right)** we modify π to be less "symmetric", setting $\alpha_1 = (16, 16), \alpha_2 = (64, 64), \alpha_3 = (32, 64), \alpha_4 = (32, 16), \alpha_5 = (20, 20)$. In this case, the independent coordinates approximation exhibits a substantially larger error.

definition to $\mathbb{R}_+^d$. A sample from $\pi_k^y(\mathrm{d}x|\tilde{x})$ can be obtained by simulating this MPP, using a slightly modified version of Algorithm 1. Instead of starting with $t_0 = 0, \tilde{Z}_0 = 0$, consider a time-grid $t_k = t_{k,0} < \cdots < t_{k,N}$, and an initial value $\tilde{Z}_{t_k} = t_k \tilde{x}$. Again, $t_k \tilde{x}$ does not necessarily belong to $\mathbb{N}^d$: one solution is to round each coordinate to the nearest integer. However, note that the discretization scheme of Algorithm 1 does not explicitly require the initial state to be integer-valued. Indeed, as noted already, Proposition 2 shows that the Poisson denoiser K_t^π, for a fixed t, can be extended to $\mathbb{R}_+^d$-valued inputs. Furthermore, as explained in Appendix E.2, the learned denoiser k_t^θ is typically trained using rescaled inputs of the form $z_t/(1+t)$ to compensate for the $Z_t = \mathcal{O}(t)$ scaling of the MPP. Thus, the learned denoiser is naturally compatible with non-integer inputs: in our numerical experiments, we observed no substantial difference when keeping $t_k \tilde{x}$ instead of rounding it.

Proposition 3. *For every $t > 0$ and $z_t \in \mathbb{N}^d$, $\pi_t(\mathrm{d}x|z_t) \propto \exp\left(-t \mathscr{D}_{\mathrm{e}}(z_t/t, x)\right) \pi(\mathrm{d}x)$.*

Proof. As by definition $\mathscr{D}_\phi(x, y) = \sum_{i=1}^d x_i \log(x_i/y_i) - x_i + y_i$, we have

$$\begin{aligned}
\exp\left(-t \mathscr{D}_\phi(t z_t, x)\right) &= \exp\left(-t \sum_{i=1}^d \frac{z_t^i}{t} \log\left(\frac{z_t^i/t}{x_i}\right) - \frac{z_t^i}{t} + x_i\right) \\
&= \exp\left(\sum_{i=1}^d z_t^i \log(x_i) - t x_i\right) \times C(z_t) \\
&\propto \prod_{i=1}^d \mathrm{e}^{-t x_i} x_i^{z_t^i} \,,
\end{aligned}$$

where $C(z_t)$ does not depend on x. We recognize the expression of $\pi_t(\mathrm{d}x|z_t)$ (2). □

Likelihood step. The likelihood step, *i.e.*, sampling from $\pi_k^y(\mathrm{d}\tilde{x}|x)$, requires an MCMC procedure. To be precise, from (7), (4), we obtain

$$\pi_k^y(\mathrm{d}\tilde{x}|x) \propto g(y|\tilde{x}) \exp\left(-t_k \sum_{i=1}^d \tilde{x}_i \log(\tilde{x}_i/x_i) - \tilde{x}_i\right) \mathrm{d}\tilde{x} =: q_y^k(\tilde{x}|x) \mathrm{d}\tilde{x} \,,$$

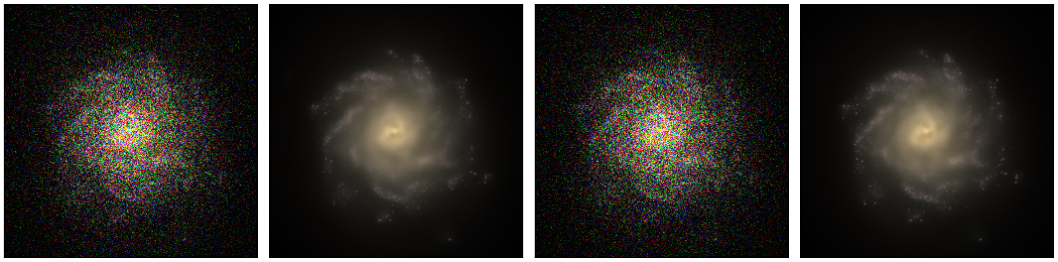

Figure 8: Same setting as Figures 5 and 6. Successive likelihood and prior steps (images 1,3 and 2,4, respectively) of MPDM-based PnP-DM.

Algorithm 6 MPDM adaptation of PnP-DM

Require: Learned denoiser $k^\theta \approx K^\pi$, likelihood $g(y|\cdot)$, measurement y, schedule $t_0 < t_1 < \cdots < t_N$; optional: $\eta > 0$, default $\eta = 1$.

1: Initialize x
2: **for** $k = 0, \dots, N-1$ **do**
3: Sample $\tilde{x}$ from $\pi_k^y(\mathrm{d}\tilde{x}|x) \propto g(y|\tilde{x})^\eta \exp\left(-t_k \mathscr{D}_{\mathrm{e}}(\tilde{x}, x)\right) \mathrm{d}\tilde{x}$ with Algorithm 7
where $\mathscr{D}_{\mathrm{e}}(\tilde{x}, x) = \sum_{i=1}^d \tilde{x}_i \log(\tilde{x}_i/x_i) - \tilde{x}_i + x_i$ ▷ Likelihood step
4: Sample x from $\pi_k^y(\mathrm{d}x|\tilde{x})$ by simulating the counting process with intensity k^θ
with initial state $t_k\tilde{x}$ at time t_k; see Algorithm 1 ▷ Prior step
5: **end for**
6: **return** x

and the associated score (gradient log with respect to $\tilde{x}$) is

$$\nabla \log q_y^k(\tilde{x}|x) = \nabla \log g(y|\tilde{x}) - t_k \sum_{i=1}^d (\log(\tilde{x}_i) - \log(x_i)) e_i \ ,$$

with $e_i = (\delta_{i,j})_{1 \leqslant j \leqslant d}$. However, this score is not well behaved when $\tilde{x}$ has coordinates close to zero. Furthermore, we need to enforce the constraints $\tilde{x} \in \mathbb{R}_+^d$. We address these points by relying on a Riemannian Langevin MCMC method; see (Girolami & Calderhead, 2011, Section 5). More precisely, we consider the manifold $(0, +\infty)^d$ equipped with the Riemannian $\mathbf{g}$ defined as $\mathbf{g}_x = \mathrm{diag}(1/x_1, \dots, 1/x_d)$. This metric is actually the *Fisher information matrix* associated with the Poisson distribution; see (Girolami & Calderhead, 2011, Section 4). With this metric, the Langevin SDE has the form

$$\mathrm{d}\tilde{X}_t = \left(\tilde{X}_t \odot \nabla \log q_y^k(\tilde{X}_t|x) + 1\right) \mathrm{d}t + \sqrt{2\tilde{X}_t} \odot \mathrm{d}B_t \ ,$$

with $\odot$ the coordinate-wise product and $1 = (1, \dots, 1) \in \mathbb{R}^d$. The Euler discretization of this SDE yields Algorithm 7. In particular, the j-th Langevin MCMC update has the form

$$\tilde{x}^{j+1} = \tilde{x}^j + \delta \times \tilde{x}^j \odot \nabla \log g(y|\tilde{x}^j) - t_k \delta \times \sum_{i=1}^d \tilde{x}_i^j (\log(\tilde{x}_i^j) - \log(x_i)) e_i + \delta \times 1 + \sqrt{2\delta \tilde{x}^j} \odot G_j \ ,$$

with $\delta > 0$ denoting the step-size and $G_j \sim \mathrm{N}(0, \mathrm{I}_d)$. Note that $\tilde{x}_i^j \log(\tilde{x}_i^j)$ is well behaved near zero.

While the Riemannian Langevin SDE is intrinsically defined on $(0, +\infty)^d$, its discretization may lead to negative outputs, especially if a large step-size δ is used; in practice, we clamp each update to guarantee positivity. Finally, Algorithm 6 includes an optional hyperparameter η, which is not present in the original PnP-DM algorithm. This addition is justified by a similar hyperparameter within the original DAPS algorithm; see below.

Algorithm 7 Riemannian Langevin MCMC algorithm on $\mathbb{R}^d_+$

Require: Score function $s : \mathbb{R}^d_+ \to \mathbb{R}^d$, time-step $\delta > 0$, number of iterations N, initial state $x_0 \in \mathbb{R}^d_+$.

1: **for** $k = 0, \dots, N-1$ **do**
2: Sample $G_k \sim \mathrm{N}(0, \mathrm{I}_d)$
3: Set $x_{k+1} = x_k + \delta x_k \odot s(x_k) + \delta \mathbf{1} + \sqrt{2\delta x_k} \odot G_k$, with $\odot$ the component-wise product
4: Set $x_{k+1} = (x_{k+1})_+$ ▷ Enforce positivity
5: **end for**
6: **return** x_N

Algorithm 8 MPDM adaptation of DAPS

Require: Learned denoiser $k^\theta \approx K^\pi$, likelihood $g(y|\cdot)$, measurement y, schedule $t_0 < t_1 < \cdots < t_N$; optional: $\eta > 0$, default $\eta = 1$.

1: Initialize x
2: **for** $k = 0, \dots, N-1$ **do**
3: Sample $z_k \sim \mathrm{Pois}(t_k x)$ ▷ Noising step
4: Compute $\hat{x} = k^\theta_{t_k}(z_k)$ (or simulate the counting process with intensity k^θ with initial state z_k at time t_k with Algorithm 1) ▷ Prior step
5: Sample x from the density $g(y|x)^\eta \mathbf{\Gamma}(x; /r^{-2}_{t_k}\hat{x}, r^{-2}_{t_k}1)\mathrm{d}x$ with Algorithm 7 ▷ Likelihood step
6: **end for**
7: **return** x

D.5 MPDM-BASED DAPS

Our MPDM version of DAPS (Zhang et al., 2025) is closely related to the original Gaussian diffusion version, described in Appendix C. We introduce, given a time-grid $t_0 < t_1 < \dots t_N$,

$$\pi^y_{t_k}(\mathrm{d}x|z_{t_k}) \propto g(y|x)\pi_{t_k}(\mathrm{d}x|z_{t_k}) \ .$$

This is mostly the same equation as (24), but here $\pi_t(\mathrm{d}x|z_t)$ is the conditional distribution associated with the MPP, defined in (2).

Prior step. Instead of a Gaussian approximation, we use a Gamma one:

$$\pi_t(\mathrm{d}x|z_t) \approx \mathbf{\Gamma}(r_t^{-2}\hat{x}(z_t), r_t^{-2}1) \ ,$$

where $\hat{x}(z_t)$ is either obtained by simulation of the MPP from z_t onward, or directly as $K^\pi_t(z_t)$. This latter choice is actually exact when π is Gamma distributed.

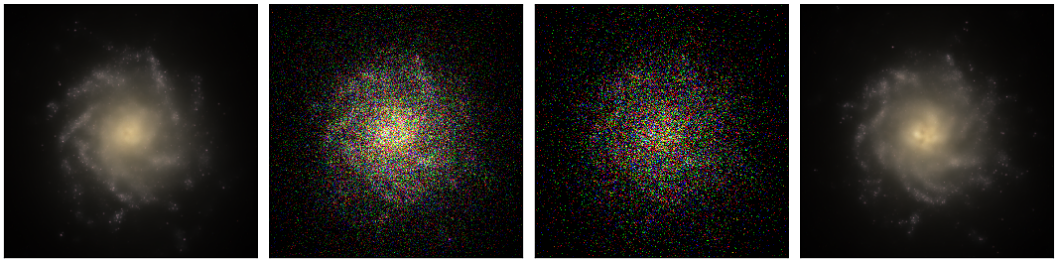

Figure 9: Same setting as Figures 5, 6 and 8. Successive prior steps (images 1,4), likelihood step (image 2), and noising step (image 3) for MPDM-based DAPS. For the noising step, we rescale by a factor of $1/(1 + t_k)$.

Example 4. *Note that when* $\pi = \gamma \sim \mathbf{\Gamma}(\alpha, \beta 1), \alpha \in (0, +\infty)^d, \beta > 0$*, from Appendix B.4,* (21)*,*

$$\pi_t(\mathrm{d}x|z_t) \sim \mathbf{\Gamma}(z + \alpha, (t + \beta)1) \ .$$

Since $K_t^\gamma(z_t) = (z_t + \alpha)/(t + \beta)$*, we have* $\pi_t(\mathrm{d}x|z_t) \sim \mathbf{\Gamma}((t + \beta)K_t^\pi(z_t), (t + \beta)1)$*, matching our "DAPS-like" approximation with* $r_t^2 = 1/(t + \beta)$*.*

Likelihood step. Similar to our PnP-DM adaptation, we use Algorithm 7 to sample from

$$\pi_{t_k}^y(\mathrm{d}x|z_{t_k}) \propto g(y|x)\mathbf{\Gamma}(r_t^{-2}\hat{x}(z_t), r_t^{-2}1) \ .$$

The Riemannian metric discussed in the previous section combines well with the Gamma density: the j-th Riemannian Langevin MCMC update is, after simplification,

$$x^{j+1} = x^j + \delta \times x^j \odot \nabla \log g(y|x^j) + \delta \sum_{i=1}^d x_i^j \times \left((r_{t_k}^{-2}\hat{x}_i^j - 1)/x_i^j - r_{t_k}^{-2} \right) e_i + \delta 1 + \sqrt{2\delta x^j} \odot G_j$$

$$= x^j + \delta \times x^j \odot \nabla \log g(y|x^j) + \delta \sum_{i=1}^d r_{t_k}^{-2} \left(\hat{x}_i^j - x_i^j \right) e_i + \sqrt{2\delta x^j} \odot G_j \ .$$

The otherwise problematic $1/x_i^j$ factors, which follow from the score of the Gamma distribution, are cancelled out.

Noising step. In analogy with the Gaussian diffusion case, we then sample $z_{t_{k+1}} \sim \mathrm{Pois}(t_{k+1}x)$. Unlike our adaptation of PnP-DM, this extra step ensures that the successive z_{t_k} lie in $\mathbb{N}^d$.

To improve empirical performance, the original DAPS includes a hyperparameter to control the weight of the likelihood g in the MCMC sampling phases, see (Zhang et al., 2025, Appendix F.2). We adopt the same strategy: in the Langevin update of Algorithm 8, we replace $\nabla \log g(y|\cdot)$ with $\eta \nabla \log g(y|\cdot)$, $\eta > 0$ (equivalently, replace $g(y|\cdot)$ with $g(y|\cdot)^\eta$). We apply the same modification to the Langevin update of PnP-DM, as we also observe better empirical performance in our inverse problem experiments when $\eta > 1$. While it tends to improve the quality of the reconstructed images, note that setting $\eta > 1$ alters the posterior distribution that we aim to sample from. In particular, it can lead to more concentrated samples, hence decreasing the posterior diversity. The values of η used in our experiments are empirically tuned and reported in Table 10.

E Numerical experiments: details

E.1 Datasets

PET. The public *BrainWeb* dataset (Aubert-Broche et al., 2006; Schramm, 2021) contains 21 PET-like brain volumes; we use 20 for training and the remaining one as validation for the inverse problems. From each volume, we extract 60 slices (the middle-most ones), resulting in 1200 256×256 training images (and 60 for validation) with four different pixel values (lowest is 0, highest is 1). During training, we add random lesions to slices [2]. We augment the dataset during training by applying random translation (± 10 pixels), horizontal flip, and small rotation ($\pm$ 2°).

Astrophysics. We downloaded the public mock galaxy images from the *EAGLE* simulation (McAlpine et al., 2016) after requesting access to the database[3]. We extracted mock images from the `RefL0100N1504_Subhalo` with `GalaxyID` $< 2.10^7$, giving 41372 RGB 256×256 images. To enforce a relative homogeneity of the training dataset, we used PCA with 100 components followed by a k-means clustering with 3 clusters, and selected one cluster. We kept 100 images for validation and 7787 for training. We augment the dataset during training by applying random translation (± 10 pixels), horizontal and vertical flips, and a random rotation. As for the PET dataset, pixel values span $[0, 1]$.

[2] https://pet-mr.github.io/brainweb/

[3] https://astro.dur.ac.uk/ICC/EAGLE/database.php

E.2 TRAINING DETAILS

While most of the literature relies on pre-trained models to evaluate Gaussian diffusion-based posterior sampling algorithms, our approach requires training models from scratch. The input z of the Poisson denoiser, $(z, t) \mapsto K_t^\pi(z)$, scales with t due to the nature of MPP. Thus, we use rescaled inputs for the network approximating K^π, denoted by k^θ, namely $z/(1+t)$ and $1/(1+t)$ instead of z, t. These design choices are similar to those of Bhattacharya et al. (2025). The network directly predicts the denoised image, rather than a residual correction to the input. We used the same strategy to train Gaussian denoisers approximating D^π. The Gaussian diffusion formulation we considered, detailed in Section 2.2, is admittedly not the most common; however, it shares several similarities with MPDM: SNR increasing with t, process $\mathbf{X}$ scaling as $X_t \approx t$, and such that $X_t/t \to X \sim \pi$ almost surely.

Architectures. For the toy example, we used a time-conditioned residual multilayer perceptron with a hidden dimension of 128 and three residual blocks. The two-dimensional input was first projected to the hidden dimension, and the diffusion time variable was encoded using a 64-dimensional time embedding followed by a two-layer MLP. Each residual block consisted of two linear layers with SiLU activations and was conditioned on the time embedding through feature-wise linear modulation (FiLM), implemented as a learned scale and shift. A final linear layer mapped the hidden representation back to the two-dimensional output space, followed by a Softplus activation to enforce non-negativity.

For the BrainWeb and EAGLE datasets, we used a U-Net architecture (Ronneberger et al., 2015) with 32 input channels, base width 256, and 5 resolution levels. The number of channels doubles at every level, with spatial downsampling by a factor of two. Each encoder and decoder level consists of two residual blocks. The residual blocks use Group Normalization and SiLU activations and are conditioned on the diffusion time through the same feature-wise linear modulation (FiLM) mechanism. The time variable is first mapped to a 128-dimensional sinusoidal embedding, followed by a two-layer MLP. The bottleneck consists of two residual blocks and contains no attention. The decoder uses nearest-neighbor upsampling followed by convolution and concatenates the corresponding encoder features through standard U-Net skip connections. A final 1×1 convolution produces the output channels, followed by a Softplus activation to enforce non-negativity.

Training. We trained our models with the Adam optimizer and a 10^{-4} learning rate. During training, each image is noised at a level t with a log-uniform distribution: $\log t$ is uniform on $[-7, 7]$. This non-uniform distribution for t is compensated with a Jacobian factor; see Appendix D.1. We use the usual l^2 loss instead of the Bregman divergence (4): after comparing both choices, we observed no significant difference in our experiments.

For the image datasets, we used a batch size of 32 and trained for 3500 epochs for the PET dataset and 1000 epochs for the astrophysics dataset. For the toy example, we used approximately 10^5 optimization steps, with batches of 1024 fresh samples from the Gamma mixture distribution defined in Appendix E.3.

E.3 INVERSE PROBLEMS: SETTINGS AND DETAILED RESULTS

Computational budget and hyperparameters. PDPS, DPS, PnP-DM and DAPS all require a time grid $t_0 < t_1 < \cdots < t_N$ to approximately simulate the underlying Gaussian diffusion/MPDM. PDPS and DPS use one NFE (denoiser call) per iteration, hence N NFEs in total; to be precise, DPS requires vector-Jacobian computations. For DAPS and PnP-DM, there is an additional inner loop at each prior step: if we fix the total number of NFEs, we must allocate them between the outer loop on $t_0 < t_1 < \dots t_N$ and the inner loop that simulates the diffusion from t onward. For fairness and simplicity, we use only one NFE for each prior step, and keep the same grid $t_0 < t_1 < \cdots < t_N$ for every algorithm. This design choice is present in DAPS: the authors observe that more NFEs in the inner loop lead to slightly better reconstruction metrics for linear inverse problems; see (Zhang et al., 2025, Appendix I). On the other hand, using only one inner-loop NFE is admittedly a slight modification of the original PnP-DM design.

In the likelihood step of DAPS and PnP-DM, we set the number of Langevin steps to 128 and the step-size to $\delta = 0.001$. While computationally cheap, this likelihood step is repeated at each

iteration, thus N times. For simplicity, we choose to only match the number of NFEs between these methods and DPS/PDPS, ignoring these Langevin steps. Note that for Gaussian diffusion-based PnP-DM and DAPS, we use a standard Langevin MCMC scheme and clamp updates that have negative values, as the authors do not discuss the case where the density is supported on $\mathbb{R}_+^d$. Meanwhile, our MPDM-based adaptations rely on Riemannian Langevin MCMC (Algorithm 7).

The hyperparameter η mimics the empirical tuning of β_y present in DAPS (Zhang et al., 2025, Appendix F.2): it effectively increases the weight of the likelihood term in the Langevin scheme. As in the original DAPS paper, we observe better reconstruction metrics when $\eta > 1$, so we keep this design choice in our implementations. This additional parameter is not present in PnP-DM: we include it to ensure fairness, as we also observe better empirical performance with $\eta > 1$. In each experiment, we test several values of η for both DAPS and PnP-DM, ranging from 1 to 20, and select the one that yields the best reconstruction metrics; see Table 10. For DAPS algorithms (either MPDM or diffusion-based), we used $r_t^2 = 1/(1+t)$. We mostly rely on continuous derivative approximation for MPDM-based DPS (see Appendix D.3), except for the EAGLE dataset (both Poisson denoising and blur problems) and the toy dataset, where we use the independent coordinates approximation until $t = 0.05$, before switching to the continuous approximation. We observed that it slightly improved the reconstruction performance in these cases.

Overall, comparison of these different methods is not straightforward, especially because DAPS and PnP-DM prior steps and likelihood steps require specific computational budget and several hyperparameters: we choose to tune η, but to keep the Langevin parameters identical for every experiment. Note that, while limited to the $\mathcal{A} = a\mathrm{I}_d$ case, PDPS requires few hyperparameters (only the time-grid design), only one NFE per iteration, and no vector-Jacobian computation.

Toy example. We consider the bidimensional Gamma mixture distribution $\pi = \frac{1}{5}\sum_{i=1}^{5} \mathbf{\Gamma}(\alpha_i, \beta_i)$ with $\alpha_1 = (16,16), \alpha_2 = (64,64), \alpha_3 = (16,64), \alpha_4 = (64,16), \alpha_5 = (20,20), \beta_1 = \beta_2 = \beta_3 = \beta_4 = (16,16), \beta_5 = (8,8)$. The MPP is completely tractable in this case; see Appendix B.4.

We sample 10 points $x_1, \dots x_{10}$ from π and draw $y_i \sim \mathrm{Pois}(ax_i)$ in both the low-count regime ($a = 0.6$) and high-count regime ($a = 3$). All algorithms use the same time grid: $s_1, \dots s_N$ are log-uniform on $[0.1, 100]$ and we set $t_i = 0.0001 + s_i - 0.1$; we choose $N = 1024$. For PnP-DM and DAPS methods, we use only one denoising step at each time t_i to ensure that all methods have identical NFE; furthermore, we keep $\eta = 1$ in order to preserve the correct target distribution. For $i = 1, \dots, 10$, every method generates 10000 approximate π^{y_i} samples: we compute the 2-Wasserstein distance against a set of 10000 exact π^{y_i} samples, and average the results over i. We repeat this procedure 10 times to obtain confidence intervals. We also compute the 2-Wasserstein distance between two distinct sets of exact samples to obtain a baseline. The full results are given in Table 4. We also generate approximate π samples with both the MPDM and the diffusion

Methods		Toy (LC)	Toy (HC)
		2-Wasserstein ↓	2-Wasserstein ↓
MPDM-based Poisson	DAPS	0.3320 ± 0.0055	0.3173 ± 0.0042
	DPS	0.2332 ± 0.0091	0.3106 ± 0.0039
	PnP-DM	1.0761 ± 0.0053	0.3749 ± 0.0037
	PDPS	0.0945 ± 0.0053	0.0732 ± 0.0023
Diffusion-based Poisson	DAPS	0.2898 ± 0.0043	0.2576 ± 0.0044
	DPS	0.2134 ± 0.0046	0.2999 ± 0.0038
	PnP-DM	1.0783 ± 0.0030	0.3553 ± 0.0030
Diffusion-based Gaussian-approximated	DAPS	0.4292 ± 0.0081	0.3241 ± 0.0039
	DPS	0.6285 ± 0.0052	0.2849 ± 0.0040
	PnP-DM	1.0792 ± 0.0072	0.3551 ± 0.0041
PDPS (exact denoiser)		0.0837 ± 0.0047	0.0625 ± 0.0016
Ground truth		0.0721 ± 0.0052	0.0469 ± 0.0022

Table 4: Average posterior sampling performance for the toy example. PDPS (exact denoiser) uses the analytical expression of the Poisson denoiser (see Appendix B.4) instead of a learned model. As a baseline, "Ground truth" indicates the 2-Wasserstein distance between two sets of exact samples.

model, using the same time-grid (with either $N = 64$ or $N = 1024$ time steps). Table 5 shows

the 2-Wassertein distance against exact π samples. Note that both models have comparable generative performance, suggesting that the posterior sampling discrepancy observed in Table 4 is due to posterior sampling algorithms rather than the quality of the learned prior.

Generative model	$N = 64$	$N = 1024$
MPDM	0.1108 ± 0.0205	0.1011 ± 0.0128
Diffusion	0.1153 ± 0.0125	0.1078 ± 0.0167
MPDM (exact denoiser)	0.1176 ± 0.0137	0.1013 ± 0.0227
Ground truth	0.0861 ± 0.0164	

Table 5: Evaluation of generative models for the toy example. 2-Wasserstein distance between exact and approximate samples.

For each imaging inverse problem, we draw an observation y for each image x in the validation dataset, compute the PSNR, SSIM, and LPIPS of the reconstructed image obtained with each posterior sampling algorithm, and average the metrics over the validation dataset. We repeat this procedure 4 times to obtain confidence intervals.

Methods		**BrainWeb denoising (LC)**			**BrainWeb denoising (HC)**		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MPDM-based Poisson	DAPS	16.72 ± 0.07	0.7570 ± 0.0025	0.1654 ± 0.0067	20.62 ± 0.08	0.8559 ± 0.0019	0.1159 ± 0.0049
	DPS	16.91 ± 0.15	0.7574 ± 0.0092	0.1429 ± 0.0110	20.49 ± 0.13	0.8477 ± 0.0086	0.1207 ± 0.0110
	PnP	16.65 ± 0.10	0.6278 ± 0.0051	0.2258 ± 0.0070	21.72 ± 0.07	0.8171 ± 0.0038	0.1408 ± 0.0071
	PDPS	16.90 ± 0.09	0.7812 ± 0.0033	0.1124 ± 0.0063	21.31 ± 0.10	0.9019 ± 0.0020	0.0585 ± 0.0038
Diffusion-based Poisson	DAPS	17.12 ± 0.07	0.7794 ± 0.0028	0.1141 ± 0.0051	21.41 ± 0.08	0.8796 ± 0.0019	0.0668 ± 0.0033
	DPS	17.22 ± 0.11	0.7915 ± 0.0032	0.0890 ± 0.0043	20.62 ± 0.14	0.8814 ± 0.0028	0.0591 ± 0.0052
	PnP	16.77 ± 0.06	0.7815 ± 0.0022	0.1113 ± 0.0042	20.91 ± 0.09	0.8842 ± 0.0017	0.0687 ± 0.0030
Diffusion-based Gaussian-approximated	DAPS	15.63 ± 0.07	0.7333 ± 0.0036	0.1821 ± 0.0073	21.03 ± 0.09	0.8753 ± 0.0020	0.0606 ± 0.0039
	DPS	15.99 ± 0.12	0.7516 ± 0.0046	0.1006 ± 0.0063	17.35 ± 0.11	0.7959 ± 0.0031	0.0859 ± 0.0050
	PnP	15.10 ± 0.01	0.7262 ± 0.0007	0.3029 ± 0.0038	20.36 ± 0.07	0.8750 ± 0.0012	0.0937 ± 0.0033

Table 6: Poisson denoising problem ($\mathcal{A} = a\mathrm{I}_d$) with the BrainWeb dataset, in the low-count ($a = 1$) and high-count ($a = 5$) regime. "Poisson" and "Gaussian-approximated" refer to which likelihood is used, similar to the suffix -p, -g of the smaller tables: either the exact Poisson likelihood or $g(y|x) \approx \mathrm{N}(\mathcal{A}(x), \mathrm{diag}(\mathcal{A}(x))$.

Methods		**BrainWeb PET (LC)**			**BrainWeb PET (HC)**		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MPDM-based Poisson	DAPS	15.82 ± 0.07	0.7296 ± 0.0028	0.1619 ± 0.0065	17.25 ± 0.06	0.7783 ± 0.0024	0.1541 ± 0.0060
	DPS	15.56 ± 0.13	0.7084 ± 0.0085	0.1395 ± 0.0096	16.84 ± 0.16	0.7503 ± 0.0100	0.1318 ± 0.0092
	PnP	16.41 ± 0.05	0.7329 ± 0.0019	0.1051 ± 0.0040	18.11 ± 0.05	0.7940 ± 0.0018	0.0828 ± 0.0031
Diffusion-based Poisson	DAPS	16.07 ± 0.06	0.7327 ± 0.0032	0.1330 ± 0.0066	17.63 ± 0.06	0.7923 ± 0.0022	0.1027 ± 0.0045
	DPS	15.82 ± 0.13	0.7401 ± 0.0042	0.0968 ± 0.0051	16.66 ± 0.20	0.7758 ± 0.0063	0.0947 ± 0.0064
	PnP	15.74 ± 0.05	0.7194 ± 0.0022	0.1755 ± 0.0056	17.33 ± 0.06	0.7819 ± 0.0023	0.1365 ± 0.0049
Diffusion-base Gaussian-approximated	DAPS	16.13 ± 0.06	0.7478 ± 0.0025	0.1111 ± 0.0058	17.60 ± 0.08	0.7930 ± 0.0046	0.1115 ± 0.0123
	DPS	14.75 ± 0.15	0.6891 ± 0.0057	0.1678 ± 0.0106	15.73 ± 0.13	0.7372 ± 0.0047	0.0962 ± 0.0059
	PnP	15.86 ± 0.06	0.7392 ± 0.0028	0.1388 ± 0.0090	17.29 ± 0.07	0.7869 ± 0.0039	0.1390 ± 0.0106

Table 7: PET (with the BrainWeb dataset), in the low-count ($a = 20$) and high-count ($a = 100$) regimes.

Methods		EAGLE denoising (LC)			EAGLE denoising (HC)		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MPDM-based Poisson	DAPS	29.97 ± 0.09	0.5871 ± 0.0036	0.3065 ± 0.0105	32.94 ± 0.06	0.6964 ± 0.0023	0.1283 ± 0.0066
	DPS	31.55 ± 0.19	0.7472 ± 0.0082	0.1783 ± 0.0152	33.16 ± 0.11	0.7748 ± 0.0044	0.1369 ± 0.0128
	PnP	26.25 ± 0.14	0.5933 ± 0.0040	0.4149 ± 0.0106	34.80 ± 0.06	0.8401 ± 0.0012	0.0991 ± 0.0040
	PDPS	32.34 ± 0.15	0.7916 ± 0.0034	0.1543 ± 0.0107	34.94 ± 0.08	0.8435 ± 0.0013	0.0863 ± 0.0044
Diffusion-based Poisson	DAPS	29.92 ± 0.09	0.7268 ± 0.0028	0.2547 ± 0.0108	34.60 ± 0.09	0.8388 ± 0.0016	0.1546 ± 0.0053
	DPS	32.46 ± 0.14	0.8060 ± 0.0026	0.2213 ± 0.0127	34.69 ± 0.09	0.8409 ± 0.0014	0.1612 ± 0.0065
	PnP	26.95 ± 0.03	0.6884 ± 0.0014	0.4085 ± 0.0053	32.99 ± 0.04	0.8203 ± 0.0009	0.1902 ± 0.0033
Diffusion-based Gaussian-approximated	DAPS	26.48 ± 0.06	0.6530 ± 0.0055	0.2322 ± 0.0086	34.66 ± 0.08	0.8372 ± 0.0016	0.1630 ± 0.0072
	DPS	31.23 ± 0.19	0.7503 ± 0.0044	0.3001 ± 0.0160	32.57 ± 0.15	0.8075 ± 0.0027	0.2122 ± 0.0120
	PnP	24.07 ± 0.02	0.6610 ± 0.0010	0.3753 ± 0.0047	33.42 ± 0.07	0.8286 ± 0.0013	0.1680 ± 0.0043

Table 8: Poisson denoising problem ($\mathcal{A} = a\mathrm{I}_d$) for the EAGLE dataset, in the low-count ($a = 4$) and high-count ($a = 20$) regime.

Methods		EAGLE blur (LC)			EAGLE blur (HC)		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MPDM-based Poisson	DAPS	27.34 ± 0.22	0.6174 ± 0.0044	0.4528 ± 0.0100	30.63 ± 0.12	0.6832 ± 0.0029	0.2970 ± 0.0105
	DPS	31.13 ± 0.22	0.7638 ± 0.0044	0.2049 ± 0.0171	31.64 ± 0.19	0.7764 ± 0.0033	0.1858 ± 0.0138
	PnP	29.83 ± 0.10	0.7341 ± 0.0023	0.3039 ± 0.0101	33.22 ± 0.05	0.8000 ± 0.0009	0.2511 ± 0.0084
Diffusion-based Poisson	DAPS	30.68 ± 0.11	0.7419 ± 0.0024	0.2724 ± 0.0098	33.01 ± 0.09	0.8000 ± 0.0018	0.2370 ± 0.0087
	DPS	31.72 ± 0.20	0.7935 ± 0.0030	0.2432 ± 0.0155	32.56 ± 0.18	0.8017 ± 0.0026	0.2129 ± 0.0104
	PnP	27.90 ± 0.09	0.6530 ± 0.0024	0.3573 ± 0.0084	29.58 ± 0.08	0.7041 ± 0.0023	0.3103 ± 0.0085
Diffusion-based Gaussian-approximated	DAPS	31.99 ± 0.10	0.7881 ± 0.0019	0.2652 ± 0.0088	33.14 ± 0.09	0.8041 ± 0.0018	0.2365 ± 0.0087
	DPS	30.90 ± 0.22	0.7691 ± 0.0041	0.3263 ± 0.0145	32.52 ± 0.21	0.7985 ± 0.0027	0.2352 ± 0.0116
	PnP	28.64 ± 0.10	0.7136 ± 0.0023	0.3632 ± 0.0088	29.74 ± 0.08	0.7153 ± 0.0024	0.3072 ± 0.0081

Table 9: Deblurring problem (with the EAGLE dataset), in the low-count ($a = 4$) and high-count ($a = 20$) regimes.

PET. Our PET setting was constructed with `parallelproj`[4] (Schramm & Thielemans, 2024). The forward operator comprised 180 equally spaced projection angles over $0 \leqslant \theta \leqslant \pi$. With 256×256 grayscale images, the discretized Radon operator satisfies $\mathcal{A} : \mathbb{R}_+^d \to \mathbb{R}_+^{d_y}$, with $d = 256 \times 256$ and $d_y = 180 \times 180$. For each angle, projection bins were uniformly distributed radially, spanning 3.5 times the image width. The projection geometry used a source-to-detector radius of 400 pixels. The image origin was centered such that the reconstructed field of view was symmetric about the origin. See Figure 10 for an illustration. For both the PET and Poisson denoising problems, each posterior sampling algorithm relies on the same log-uniform time-grid $t_0 < \cdots < t_N$, with $t_N = 100$ and $N = 128$,

Astrophysics. We considered a two-dimensional Gaussian blurring forward operator. The input images were convolved with a normalized 65×65 Gaussian kernel with standard deviation $\sigma = 3$. pixels. Zero-padding was used at the image boundaries to preserve the image dimensions. Here, $\mathcal{A} : \mathbb{R}_+^d \to \mathbb{R}_+^d$ with $d = d_y = 3 \times 256 \times 256$. See Figure 11 for a visual example. For both the blur and Poisson denoising problems, each posterior sampling algorithm relies on the same log-uniform time-grid $t_0 < \cdots < t_N$, with $t_N = 100$ and $N = 256$.

E.4 PET: ADDITIONAL MEASUREMENTS

The classical metrics we use to evaluate posterior sampling algorithms measure the quality of a given reconstructed image. They therefore provide only a partial assessment of sampling quality, as they do not directly account for the dispersion or diversity of the generated samples. Typically, in

[4] https://parallelproj.readthedocs.io/en/stable/

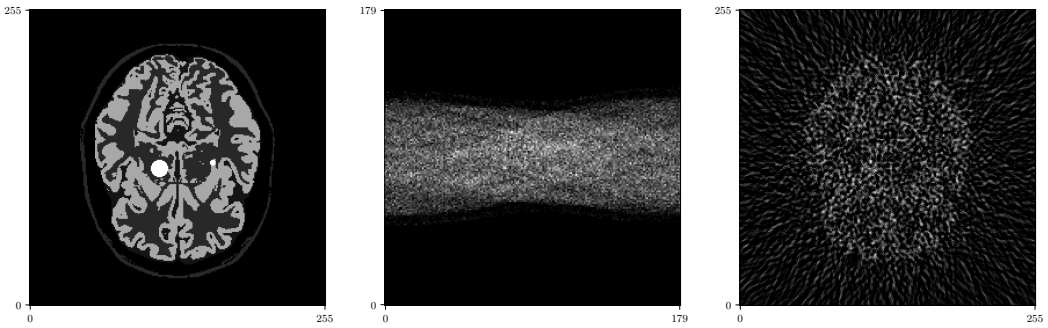

Figure 10: PET illustration. **(Left)** Ground-truth image x from the BrainWeb dataset, with lesions (bright dots). **(Center)** the *sinogram*, *i.e.*, observation $y \sim \mathrm{Pois}(\mathcal{A}(x))$. Each pixel represents the number of detections for a given pair of detectors. **(Right)** Filtered back-projection of y.

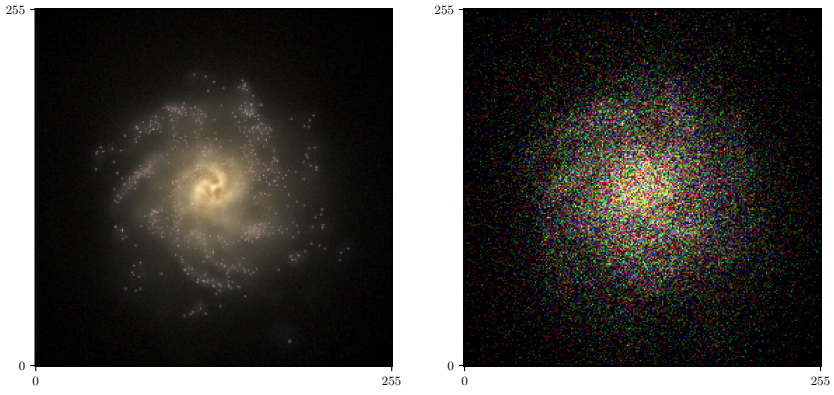

Figure 11: Gaussian blurring illustration. **(Left)** Ground-truth image x from the EAGLE dataset. **(Right)** Observation $y \sim \mathrm{Pois}(\mathcal{A}(x))$.

	Toy		BrainWeb denoising		BrainWeb PET		Eagle denoising		Eagle blur	
	LC	**HC**	**LC**	**HC**	**LC**	**HC**	**LC**	**HC**	**LC**	**HC**
PnP-DM (MPDM) η	1	1	8	5	14	8	5	4	7	5
DAPS (MPDM) η	1	1	5	4	8	5	2	3	6	3
PnP-DM-p (diffusion) η	1	1	4	2	8	4	10	10	7	3
DAPS-p (diffusion) η	1	1	6	5	12	8	5	7	8	6
PnP-DM-g (diffusion) η	1	1	9	2	6	4	9	2	4	2
DAPS-g (diffusion) η	1	1	2	2	10	8	2	2	4	5

Table 10: We specify the parameter η chosen for the algorithms relying on a Langevin MCMC scheme: our MPDM-based PnP-DM (Algorithm 6) and DAPS (Algorithm 8), as well as their diffusion counterparts (either with true Poisson likelihood, indicated by -p, or its Gaussian approximation, indicated by -g). Values of η ranging from 1 to 20 were tested in each setting, and we chose the ones leading to the overall best reconstruction metrics.

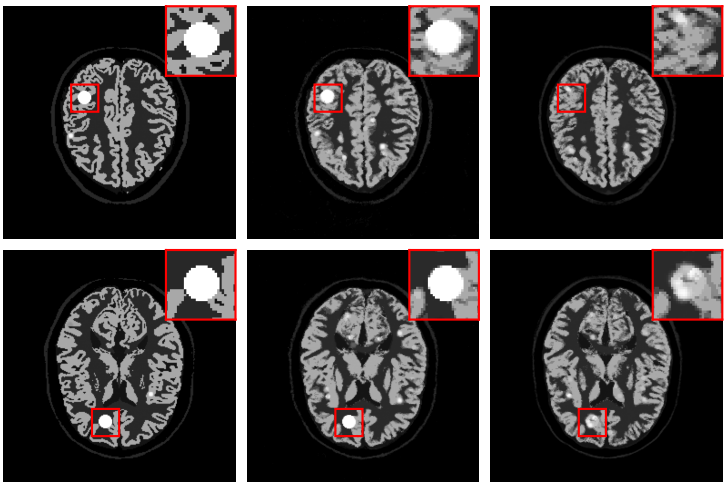

Figure 12: Poisson denoising problem with the BrainWeb dataset, in the low-count regime. **(Left)** Ground truth image from the BrainWeb validation dataset. **(Center)** Average of 4 samples obtained with PDPS. **(Right)** Average of 4 samples obtained with diffusion-based DPS.

low-count regimes, posterior sampling methods should capture the dispersion of the true posterior. Moreover, the relevant notion of reconstruction quality may depend on the downstream task and the specific inverse problem at hand. In PET, for example, detecting lesions and accurately quantifying the associated uncertainty can be more relevant than the image quality measured by the aforementioned metrics. Figures 12 and 13 illustrate the detection of lesions with the BrainWeb dataset, in low-count regimes.

To quantify this, we also evaluate reconstruction errors specifically within the lesions; see Tables 11 and 12. To do so, we compute the mean absolute error between the ground truth images and the approximate posterior samples, restricted to lesions (pixels with value > 0.9 on either image). If x denotes the ground truth and $\hat{x}$ the reconstruction, we define it as

$$l(x, \hat{x}) = \frac{1}{\#\mathcal{L}(x) \cup \mathcal{L}(\hat{x})} \sum_{i=1}^{d} |\mathbb{1}_{\mathcal{L}(x)} - \mathbb{1}_{\mathcal{L}(\hat{x})}| \,,$$

with $\mathcal{L}(x) = \{1 \leqslant i \leqslant d : x_i > 0.9\}, \mathcal{L}(\hat{x}) = \{1 \leqslant i \leqslant d : \hat{x}_i > 0.9\}$. In particular, we show that, while diffusion-based DPS is on par with, or even outperforms, MPDM-based methods on several metrics reported in Tables 6 and 7, it nevertheless exhibits larger reconstruction errors in lesion regions. This is still an image metric, however, and not a proper posterior calibration evaluation.

E.5 Generative modeling: gallery

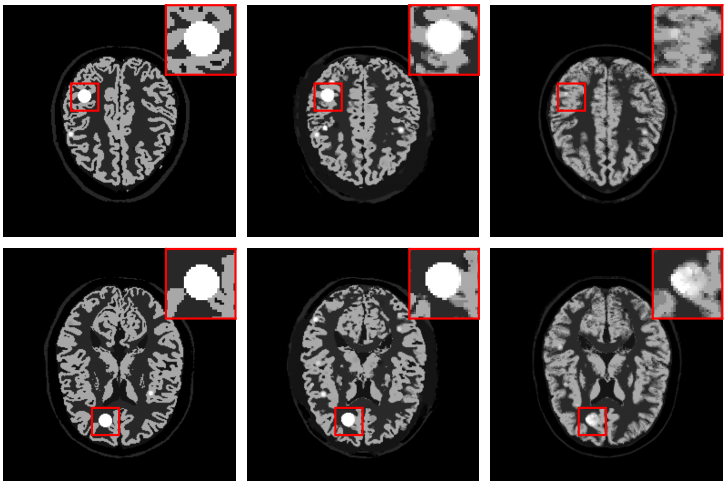

Figure 13: PET problem with the BrainWeb dataset, in the low-count regime. **(Left)** Ground truth image from the BrainWeb validation dataset. **(Center)** Average of 4 samples obtained with MPDM-based PnP-DM. **(Right)** Average of 4 samples obtained with diffusion-based DPS.

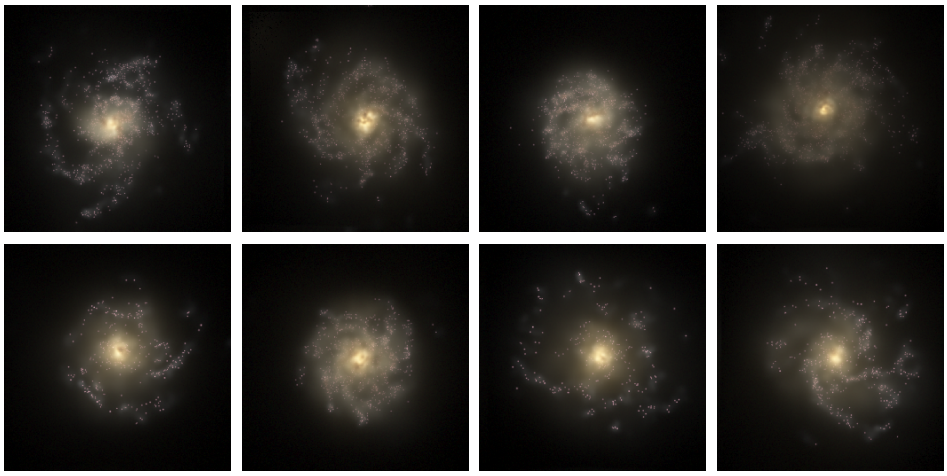

Figure 14: Images generated by our MPDM trained on the EAGLE dataset.

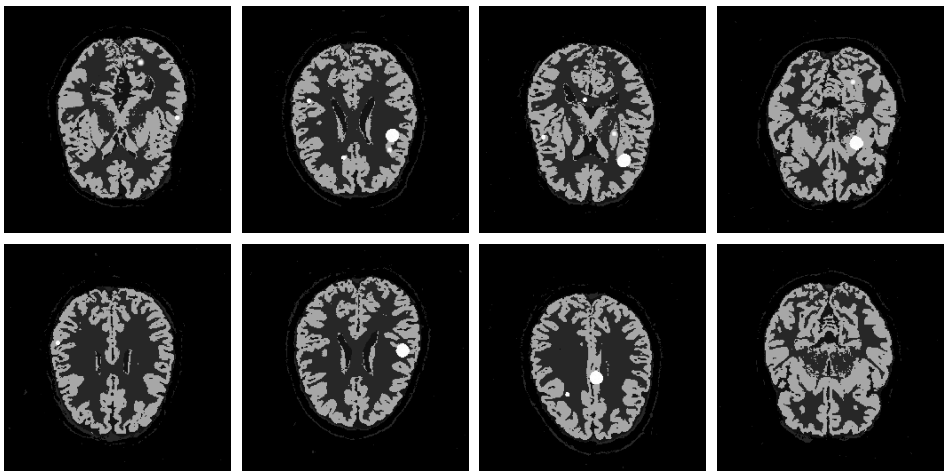

Figure 15: Images generated by our MPDM trained on the BrainWeb dataset.

Methods	BrainWeb denoising (LC)
PDPS (MPDM)	0.7264 ± 0.0051
DPS-p (diffusion)	0.7832 ± 0.0094

Table 11: Average $l(x, \hat{x})$, where x is a validation sample of the BrainWeb dataset, and $\hat{x}$ is an approximate posterior sample obtained with either diffusion-based DPS or PDPS, for the Poisson denoising problem ($\mathcal{A} = a\mathrm{I}_d$) in the low-count regime.

Methods	BrainWeb PET (LC)
PnP-DM (MPDM)	0.7573 ± 0.0080
DPS-p (diffusion)	0.8458 ± 0.0228

Table 12: Average $l(x, \hat{x})$, where x is a validation sample of the BrainWeb dataset, and $\hat{x}$ is an approximate posterior sample obtained with either diffusion-based DPS or PDPS, for the PET problem in the low-count regime.