A STATISTICAL BENCHMARK FOR DIFFUSION POSTERIOR SAMPLING ALGORITHMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a statistical benchmark for diffusion posterior sampling (DPS) algorithms in linear inverse problems. Our test signals are discretized Lévy processes whose posteriors admit efficient Gibbs methods. These Gibbs methods provide gold-standard posterior samples for direct, distribution-level comparisons with DPS algorithms. They also serve as oracle denoisers in the reverse diffusion, which enables the isolation of the error that arises from the approximations to the likelihood score. We instantiate the benchmark with the minimum-mean-squared-error optimality gap and posterior coverage tests and evaluate popular algorithms on the inverse problems of denoising, deconvolution, imputation, and reconstruction from partial Fourier measurements. We release the benchmark code at https://github.com/emblem-saying/dps-benchmark. The repository exposes simple plug-in interfaces, reference scripts, and config-driven runs so that new algorithms can be added and evaluated with minimal effort. We invite the community to contribute and report results.

1 INTRODUCTION

Diffusion models are among the leading generative models in imaging (Rombach et al., 2022), visual computing (Po et al., 2024), finance and time-series analysis (Huang et al., 2024; Rasul et al., 2021), de novo protein and drug design (Watson et al., 2023; Alakhdar et al., 2024), natural language processing (Li et al., 2022), and other domains. Their ability to model complex distributions has motivated their use as priors in the Bayesian resolution of inverse problems. In fact, reconstruction methods that leverage diffusion models are competitive or state-of-the-art in, *e.g.*, deconvolution (Ren et al., 2023), phase retrieval (Xue et al., 2025), magnetic resonance imaging and computed tomography reconstruction (Chung & Ye, 2022; Liu et al., 2023), weather-artifact removal (Özdenizci & Legenstein, 2023), task-conditioned protein design (Bogensperger et al., 2025), audio bandwidth extension and dereverberation (Lemercier et al., 2024), and denoising of financial time-series (Wang & Ventre, 2024).

This empirical success has come despite diffusion models lacking a natural mechanism for conditioning on measurements and active research explores how to incorporate the likelihood (Yismaw et al., 2025; Erbach et al., 2025). Currently, conditioning strategies are evaluated in one of two ways. (i) With respect to downstream applications: As an example, evaluations with respect to perceptual metrics such as the structural similarity (Wang et al., 2004), the Fréchet inception distance (Heusel et al., 2017), or the learned perceptual image patch similarity (Zhang et al., 2018) are common in the imaging sciences. However, as pointed out by Pierret & Galerne (2025b), these metrics are ill-suited for the statistical evaluation of posterior-sampling algorithms. (ii) In overly simplistic settings: A common fallback is to evaluate conditioning strategies in synthetic settings with (finite-component) Gaussian mixture priors. Such mixtures remain light-tailed with the tail decreasing exponentially like the widest component and, consequently, they cannot reproduce power-law-like extremes that are common in, *e.g.*, asset returns (Blattberg & Gonedes, 1974; Cont, 2001) and statistics of images (Wainwright & Simoncelli, 1999). Benchmarks built on such priors can therefore overstate posterior quality. A proper statistical evaluation in realistic settings is critical in high-stakes applications such as medical imaging, remote sensing, and finance, where decisions based on reconstructions and their associated uncertainties may have significant consequences.

1.1 CONTRIBUTIONS

We propose such a statistical benchmark for diffusion posterior sampling (DPS) algorithms for linear inverse problems. We consider a setting similar to Bohra et al. (2023) in which test signals are obtained from discretized sparse Lévy-process priors that admit efficient posterior-sampling algorithms. Indeed, they admit efficient Gibbs methods with exact conditionals that provide gold-standard posterior samples. The framework supports general posterior-level comparisons—*e.g.*, (sliced) Wasserstein or energy distances or calibration via coverage and posterior predictive checks—by furnishing matched samples obtained from the DPS algorithms and the gold-standard Gibbs methods.

We introduce a new template for DPS algorithms, where each update step utilizes *samples* from the denoising posterior—as opposed to only the minimum-mean-squared error (MMSE) point estimate—and we show how several popular DPS algorithms can be re-expressed within this template. This template arises naturally in our framework because the Gibbs methods can provide these denoising-posterior samples. This construction enables the isolation of the error that is attributable to the likelihood-score approximation by replacing the learned denoiser with an oracle MMSE denoiser computed from Gibbs samples at each reverse-diffusion step. Finally, we instantiate the framework with the MMSE optimality gap and highest posterior density coverage checks across the inverse problems of denoising, deconvolution, imputation, and reconstruction from partial Fourier measurements. The benchmark code is available in an online repository that contains efficient implementations of sampling routines and a containerized runtime that allows novel algorithms to be benchmarked easily.

1.2 RELATED WORK

For unconditional sampling, many works derive theoretical bounds on various distances between a target distribution and the distribution obtained by (approximations of) the reverse stochastic differential equation (SDE) (see section 2). For example, Gao et al. (2025) bound the Wasserstein-2 distance with respect to the discretization error of the SDE under the assumption that the target distribution is smooth and log-concave. This directly bounds the number of reverse-diffusion steps that are needed to obtain a desired accuracy. Under absolute continuity of the target with respect to a Gaussian, Strasman et al. (2025) bound the Kullback–Leibler divergence with respect to properties of the noise schedule. Additional results in other distances can be found in the references cited therein.

A common assumption that simplifies the analysis is that of a Gaussian target. In that case, many objects in the forward and reverse SDE admit closed forms, which facilitates the computation of various bounds. For example, Hurault et al. (2025) analyze the error due to the finite number of prior samples used in the estimation of the prior score (which is affine in this case) and track its propagation through the iterations of the reverse-SDE solver. Pierret & Galerne (2025b) derive explicit solutions to the SDE and use those to derive bounds on the Wasserstein-2 distance to the distributions that are obtained via Euler–Maruyama discretizations.

Works that consider conditional generation and are closest to the present paper are Pierret & Galerne (2025a) and Crafts & Villa (2025). Pierret & Galerne (2025a) derive expressions for the Wasserstein-2 distances between the conditional forward marginals and the distributions induced by specific likelihood approximations in the reverse SDE under the assumption of a Gaussian prior. In contrast, our framework handles a broader set of priors (discretized Lévy processes) and accommodates a broader set of algorithms than those that rely on specific likelihood approximations. Moreover, deriving explicit expressions for new algorithms often requires a substantial amount of nontrivial mathematics. In contrast, our benchmark is deliberately designed for a plug-and-play evaluation of novel algorithms. Crafts & Villa (2025) evaluate DPS algorithms under the assumption of a (finite-component) Gaussian mixture prior numerically. Similar to the present work, they provide reference objects to the DPS algorithms to ensure a fair evaluation. However, they only consider Gaussian mixture priors, which cannot reproduce power-law-like extremes and can overstate posterior quality.

Beyond diffusion-specific theory, Thong et al. (2024) evaluate posterior calibration by checking the coverage of credible regions produced by different Bayesian recovery strategies. They find that recovery strategies that utilize diffusion models often under-report uncertainty. A shortcoming of their approach is that they use an empirical distribution of images as a surrogate for the prior distribution. Our framework, by contrast, relies on known priors from which infinitely many signals and corresponding measurements can be generated. It isolates algorithmic errors without resorting to surrogate priors and supports fair, repeatable comparisons across tasks and algorithms.

2 BACKGROUND

Bayesian Linear Inverse Problems We seek to estimate a signal $\mathbf{x} \in \mathbb{R}^d$ from the measurements

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{1}$$

where the *forward operator* $\mathbf{A} \in \mathbb{R}^{m \times d}$ models the noiseless linear measurement acquisition and $\mathbf{n} \in \mathbb{R}^m$ is additive noise. In the Bayesian resolution of this problem (see, *e.g.*, Stuart (2010)), the signals are modeled as a random variable, denoted $\mathbf{X}$, with values in $\mathbb{R}^d$ and distribution $p_{\mathbf{X}}$, referred to as the *prior*. Given any measurement $\mathbf{y}$, the ultimate goal is to analyze the *posterior* $p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$ which is related to the *likelihood* $p_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}$ and the prior $p_{\mathbf{X}}$ via Bayes' rule, which states that

$$p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x}) \propto p_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}(\mathbf{y}) p_{\mathbf{X}}(\mathbf{x}). \tag{2}$$

In contrast to classical variational methods (see, *e.g.*, Scherzer et al. (2008)), the posterior distribution provides natural means to quantify uncertainty and can be summarized by various point estimators. We provide a precise description of point estimators that are relevant in this work in appendix A.

For a given signal $\mathbf{x}$, the likelihood $p_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}$ is determined by the distribution of the noise. A common assumption on the noise is that it is a vector of independent and identically distributed (i.i.d.) Gaussian random variables with mean zero and variance σ_{n}^2.[1] In this case, the likelihood is given by

$$p_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}(\mathbf{y}) \propto \exp\!\left(-\tfrac{1}{2\sigma_{\mathrm{n}}^2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2\right). \tag{3}$$

Thus, once the forward model and the noise distribution are specified, the remaining modeling choice is the prior. Diffusion models are good candidates due to their ability to encode complex distributions.

Diffusion Models Diffusion models were introduced by Song et al. (2021) by unifying the discrete approaches from Song & Ermon (2019) and Ho et al. (2020) in a continuous theory based on SDEs (Klenke, 2020, Chapters 25 and 26). We denote the (diffusion) SDE with *drift coefficient* $\mathbf{f} : \mathbb{R}^d \times \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ and *diffusion coefficient* $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ as

$$\mathrm{d}\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{W}_t \tag{4}$$

with some suitable initial condition $\mathbf{X}_0$, where $\mathbf{W}_t$ is the standard Wiener process. In our setup, the initial condition is the random variable that describes the signal, thus $\mathbf{X}_0 = \mathbf{X}$. Under suitable choices for $\mathbf{f}$ and g, the forward process admits a limiting marginal $\mathbf{X}_\infty$ as $t \to \infty$. Sampling from $p_{\mathbf{X}_0}$ can then proceed by simulating the SDE (4) in reverse with initial condition $\mathbf{X}_\infty$. By Anderson's theorem (Anderson, 1982), the reverse SDE that reproduces the forward marginals satisfies

$$\mathrm{d}\mathbf{X}_t = \left(\mathbf{f}(\mathbf{X}_t, t) - g^2(t)\nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t)\right)\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{W}_t, \tag{5}$$

where $p_{\mathbf{X}_t}$ denotes the density of $\mathbf{X}_t$ defined by the forward process, and $\mathrm{d}t$ is negative.

The primary challenge in this approach lies in the computation of the *scores* $\nabla \log p_{\mathbf{X}_t}$ for all $t > 0$. A fundamental relation known as Tweedie's formula connects the score with the MMSE denoiser: As we derive rigorously in appendix B, for $\mathbf{f}(\mathbf{x}, t) = -\frac{\beta(t)}{2}\mathbf{x}$ and $g(t) = \sqrt{\beta(t)}$,[2] we have that

$$\nabla \log p_{\mathbf{X}_t}(\mathbf{x}) = -\sigma(t)^{-2}\big(\mathbf{x} - \alpha(t)\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]\big), \tag{6}$$

where $\alpha(t) = \exp(-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s)$ and $\sigma^2(t) = 1 - \alpha^2(t)$. This yields a practical way of computing $\nabla \log p_{\mathbf{X}_t}(\mathbf{x})$ through the resolution of the MMSE denoising problem of finding $\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]$. In standard applications where the goal is the generation of new signals, this is typically tackled by approximating the map $(\mathbf{x}, t) \mapsto \mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]$ with a neural network that is learned in an off-line step. In our benchmark, we can instead obtain oracle MMSE denoisers via Gibbs methods and thereby eliminate approximation errors from a learned surrogate to isolate errors in DPS algorithms.

Implementing the reverse SDE for generation requires its time discretization, for instance with Euler–Maruyama techniques (Higham, 2001). In practice, researchers typically use the alternative denoising diffusion probabilistic model (DDPM) backward process (starting from $\mathrm{Gauss}(\mathbf{0}, \mathbf{I})$)

$$\mathbf{X}_{t-1} = \tfrac{1}{\sqrt{1-\beta_t}}\left(\mathbf{X}_t + \beta_t \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t)\right) + \sqrt{\beta_t}\mathbf{Z}_t, \tag{7}$$

that originates from the discrete-time Markov chain that was initially proposed by Sohl-Dickstein et al. (2015) and revisited and popularized by Ho et al. (2020). We relate it to the Euler–Maruyama discretization of the reverse SDE through Taylor expansions in appendix B.1.

[1] Our framework supports more general (possibly non-Gaussian) likelihoods, see section 3.

[2] The *variance-preserving* (VP) formulation (Song et al., 2021, Section 3.4) with standard normal limiting marginal, where $\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ controls the speed of the contraction to zero and how much noise is injected.

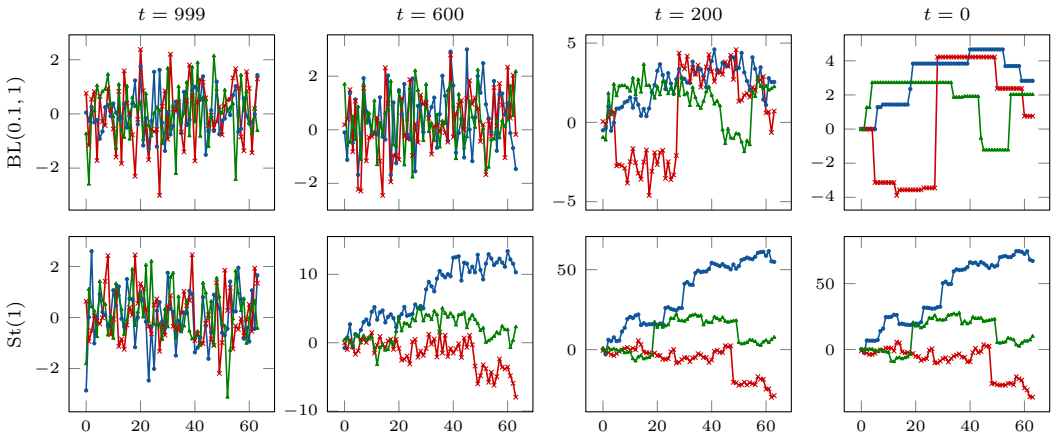

Figure 1: Unconditional reverse-diffusion trajectories obtained by DDPM using the oracle denoiser. Rows: Increment distributions. Columns: Diffusion times. Line styles: Different random states.

We show trajectories of signals generated by this backward process using the oracle MMSE denoiser in fig. 1 and motivate the oracle denoiser by showing the histograms of increments obtained by the learned denoiser versus the oracle denoiser for a $\mathrm{St}(1)$ increment target in fig. 2. The construction of the signals and the oracle denoiser are described in section 3 and our notations of various distributions are summarized in appendix C.2. The signals generated by using the oracle denoiser follow the increment distribution almost perfectly. Residual errors are due to the discretization error of the reverse diffusion and Monte Carlo error of the oracle.

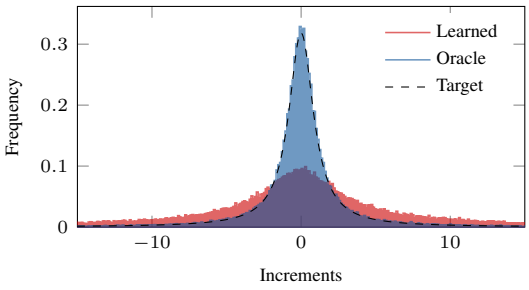

Figure 2: Histogram of increments of signals obtained by DDPM with the oracle and learned denoiser.

Diffusion Posterior Sampling The reverse-diffusion sampler from the previous section can be adapted to sample from a posterior by replacing the prior score $\nabla \log p_{\mathbf{X}_t}$ with the posterior score

$$\nabla \log p_{\mathbf{X}_t|\mathbf{Y}=\mathbf{y}} = \nabla \log p_{\mathbf{X}_t} + \nabla\big(\mathbf{x} \mapsto \log p_{\mathbf{Y}|\mathbf{X}_t=\mathbf{x}}(\mathbf{y})\big) \tag{8}$$

for some given measurement $\mathbf{y}$, obtained by Bayes' theorem. Although the dependence between $\mathbf{Y}$ and $\mathbf{X}_0$ is known through (1) and the likelihood is explicitly modeled via (3), it is generally challenging to relate $\mathbf{Y}$ and $\mathbf{X}_t$ for any $t > 0$. To overcome this, the conditioning on the measurements is usually done in one of two ways. (i) A learned component models the conditional posterior score and also gets the measurements as input. This strategy is pursued by, *e.g.*, Liu et al. (2023); Özdenizci & Legenstein (2023); Bogensperger et al. (2025); Saharia et al. (2023), and is advantageous when the measurement process is unknown, difficult to model, or prohibitively expensive to evaluate. However, reconstructions obtained by this strategy typically degrade under shifts in measurement conditions, since the learned components cannot adapt to the new measurement conditions. (ii) The Bayesian separation that is described in (8) is pursued and the likelihood score is approximated. This is done by, *e.g.*, Chung & Ye (2022); Xue et al. (2025) and the methods reviewed by Lemercier et al. (2024), and is advantageous when the measurement process is known, relatively inexpensive to evaluate, and subject to change, but prior knowledge should be reused, which is frequently the case in, *e.g.*, imaging or remote sensing applications. However, this requires approximations to the likelihood score $\nabla(\mathbf{x} \mapsto \log p_{\mathbf{Y}|\mathbf{X}_t=\mathbf{x}}(\mathbf{y}))$ for all $t > 0$.

Our benchmark can evaluate either strategy (and any other method that claims to sample from a posterior distribution). The first approach, however, relies on black-box learning of the conditional posterior score and its performance heavily depends on various implementation details. Thus, we primarily focus on the second approach that necessitates approximations of the likelihood score (and more general DPS algorithms with explicit conditioning, see our proposed generalization in section 3). Our framework can supply reference objects—posterior samples and oracle denoisers via Gibbs methods—to isolate and quantify the impact of these approximations.

3 PROPOSED FRAMEWORK

The prior distributions in our framework will be that of signals of length d obtained by regularly spaced samples of processes with independent, stationary increments (Lévy processes, described in appendix C). Let s be such a process and let the unit-step increments be $\mathbf{u}_k = s(k) - s(k-1)$ for $k = 1, \dots, d$. Independence and stationarity imply that the distribution of $\mathbf{u}_k$, the *increment distribution* p_U, does not depend on k. The increment vector is related to the signal $\mathbf{x}$ via $\mathbf{u} = \mathbf{D}\mathbf{x}$, where $\mathbf{D}$ is a finite-difference matrix with an initial condition that allows us to write $\mathbf{x} = \mathbf{D}^{-1}\mathbf{u}$ where $\mathbf{D}^{-1}$ is a lower-triangular matrix of ones. This gives a convenient way to synthesize signals once $\mathbf{u}$ is drawn. The independence of the increments implies that the density of the discrete signal is

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{k=1}^{d} p_U\big((\mathbf{D}\mathbf{x})_k\big). \tag{9}$$

We consider four increment distributions that are commonly used in sparse-process models: Gaussian, Laplace, Student-t, and Bernoulli–Laplace (spike-and-slab). Such increment distributions are sparse or heavy-tailed according to the taxonomy in (Unser & Tafti, 2014) and are relevant in signal and image processing, finance, and other fields (Schoutens, 2003). A precise definition of Lévy processes, the matrix $\mathbf{D}$, the increment distributions and their notation along with a discussion about extensions to higher-dimensional signals or signals with more complicated graph structure are given in appendix C.

Efficient Posterior Sampling With the prior distribution specified in (9) and the assumption of Gaussian noise, the posterior associated to the inverse problem (1) is

$$p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x}) \propto \exp\big(-\tfrac{1}{2\sigma_{\mathrm{n}}^2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2\big)p_{\mathbf{X}}(\mathbf{x}) = \exp\big(-\tfrac{1}{2\sigma_{\mathrm{n}}^2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2\big)\prod_{k=1}^{d} p_U\big((\mathbf{D}\mathbf{x})_k\big). \tag{10}$$

Unless p_U is a Gaussian (the simplified setting in Pierret & Galerne (2025b)), this posterior is not conjugate, so neither closed-form sampling nor direct evaluation of moments is available. Nevertheless, for the increment distributions used in this paper, the posterior distributions admit efficient Gibbs methods via standard latent-variable augmentations. Motivation and more details about the Gibbs methods, such as the burn-in period B and the number of samples S, are provided in appendix D.

The Gaussian, Laplace, and Student-t distributions admit latent representations as infinite-component Gaussian mixtures, which makes them suitable for the Gaussian latent machine (GLM) framework that was recently introduced by Kuric et al. (2025). The GLM framework is generally applicable to distributions of the form

Algorithm 1 GLM Gibbs method.

Require: $\mathbf{x}_0 \in \mathbb{R}^d$, $\mathbf{K} \in \mathbb{R}^{n\times d}$, conditional latent distributions $\{p_{\mathbf{Z}_i|X}\}_{i=1}^n$ and maps $\{\mu_i, \sigma_i^2\}_{i=1}^n$

1: **for** $s = 1, \dots, B+S$ **do**
2: Draw $\mathbf{z}_i \sim p_{\mathbf{Z}_i|X=(\mathbf{K}\mathbf{x}_{s-1})_i}$ ▷ *parallel over* i
3: Draw $\mathbf{x}_s \sim \mathrm{Gauss}(\boldsymbol{\mu}(\mathbf{z}), \boldsymbol{\Sigma}(\mathbf{z}))$
4: **return** $\{\mathbf{x}_{B+s}\}_{s=1}^S$

$$p(\mathbf{x}) \propto \prod_{k=1}^{n} \phi_k\big((\mathbf{K}\mathbf{x})_k\big) \tag{11}$$

where $\mathbf{K} \in \mathbb{R}^{n\times d}$ and all distributions $\phi_1, \phi_2, \dots, \phi_n : \mathbb{R} \to \mathbb{R}$ have a latent representation

$$\phi_k(t) = \int_{\mathbb{R}} g_{\mu_k(z),\sigma_k^2(z)}(t) f_k(z)\,\mathrm{d}z, \tag{12}$$

where the *latent distribution* f_i and the *latent maps* $\mu_i, \sigma_i^2 : \mathbb{R} \to \mathbb{R}$ depend on the distribution ϕ_i, and g_{μ,σ^2} is the density of a one-dimensional Gaussian distribution with mean μ and variance σ^2. The posterior distribution in (10) can be cast into this framework by rewriting it as

$$p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x}) \propto \prod_{k=1}^{m} g_{\mathbf{y}_k,\sigma_{\mathrm{n}}^2}\big((\mathbf{A}\mathbf{x})_k\big)\prod_{k=1}^{d} p_U\big((\mathbf{D}\mathbf{x})_k\big) = \prod_{k=1}^{m+d} \phi_k\big((\mathbf{K}\mathbf{x})_k\big) \tag{13}$$

by setting $\mathbf{K} = [\mathbf{A}; \mathbf{D}]$, $\phi_k = g_{\mathbf{y}_k,\sigma_{\mathrm{n}}^2}$ for $k = 1, \dots, m$, and $\phi_k = p_U$ for $k = m+1, \dots, m+d$. We summarize the GLM sampling in algorithm 1. Importantly, non-Gaussian likelihoods can be handled by adapting the first m distributions appropriately.

The introduction of an appropriate n-dimensional random variable $\mathbf{Z}$ with non-trivial distribution (see the details in Kuric et al. (2025)) enables the efficient sampling from the conditionals: Sampling $\mathbf{X} \mid \mathbf{Z} = \mathbf{z}$ amounts to sampling a Gaussian with covariance and mean

$$\boldsymbol{\Sigma}(\mathbf{z}) = (\mathbf{K}^T \boldsymbol{\Sigma}_0(\mathbf{z})^{-1} \mathbf{K})^{-1} \text{ and } \boldsymbol{\mu}(\mathbf{z}) = \boldsymbol{\Sigma}(\mathbf{z}) \mathbf{K}^T \boldsymbol{\Sigma}_0(\mathbf{z})^{-1} \boldsymbol{\mu}_0(\mathbf{z}), \tag{14}$$

respectively, where $\boldsymbol{\Sigma}_0(\mathbf{z}) = \operatorname{diag}\big(\sigma_1^2(\mathbf{z}_1), \dots, \sigma_n^2(\mathbf{z}_n)\big)$ and $\boldsymbol{\mu}_0(\mathbf{z}) = \big(\mu_1(\mathbf{z}_1), \dots, \mu_n(\mathbf{z}_n)\big)$. Sampling $\mathbf{Z} \mid \mathbf{X} = \mathbf{x}$ amounts to sampling n independent one-dimensional *conditional latent distributions* $p_{\mathbf{Z}_1 \mid X = (\mathbf{K}\mathbf{x})_1}, \dots, p_{\mathbf{Z}_n \mid X = (\mathbf{K}\mathbf{x})_n}$ that depend on the distributions $\phi_1, \dots, \phi_n$ and are given in table 3 in the appendix along with the corresponding latent distributions and latent maps.

For the Bernoulli–Laplace distribution, we adapt the algorithm proposed by Bohra et al. (2023) that introduces two d-dimensional latent variables: A Bernoulli indicator ("on"/"off") and a Laplace-distributed increment height. For a self-contained exposition, we rigorously derive the resulting Gibbs method in appendix D.1.

The Gibbs methods that we just described are suitable for the generation of the gold-standard samples from the posterior that corresponds to the initial inverse problem (1) as well as the generation of samples from the denoising posteriors in the DPS algorithms. In the latter case, the forward operator $\mathbf{A}$ is the identity, the measurements are the noisy intermediate reconstructions $\mathbf{x}_t$, and the noise variance $\sigma_{\mathrm{n}}^2 = \sigma_t^2$ follows the schedule at timestep t. When these Gibbs methods are embedded within the reverse-diffusion loop, an efficient implementation is paramount to achieve acceptable runtimes. This is especially true for Bernoulli–Laplace increments, where the sequential drawing of the binary support vector occurs inside the outer Gibbs loop, which in turn may be nested within the reverse-diffusion loop. Accordingly, we deliberately tailored our implementation—which we regard as a substantial contribution—to modern, highly parallel compute units and optimized several components, including custom `CUDA`- and `Triton`-compiled sampling routines and incremental updates based on the Woodbury–Sherman–Morrison identities (see appendix D.2).

A Generalized DPS Template Widely used methods such as diffusion plug-and-play (DPnP) (Xu & Chi, 2024), fall outside the pattern described in section 2—approximating the likelihood score inside the reverse diffusion. We therefore introduce a simple template that is natural in our setting and accommodates a broader set of DPS algorithms.

We characterize DPS algorithms as an iteration rule that can be summarized into a two-stage process: Given an iterate $\mathbf{x}_t$ with associated noise variance σ_t^2, the computation of the next iterate $\mathbf{x}_{t-1}$ is done by (i) drawing S samples denoted $\{\bar{\mathbf{x}}_s\}_{s=1}^S$ from the denoising posterior $p_{\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}_t} \propto \exp\big(-\frac{1}{2\sigma_t^2} \| \cdot - \mathbf{x}_t \|^2\big) p_{\mathbf{X}_0}(\cdot)$, and (ii) computing the next iterate $\mathbf{x}_{t-1}$ through an update step $\mathcal{S}$ that may utilize the current iterate $\mathbf{x}_t$, the samples $\{\bar{\mathbf{x}}_s\}_{s=1}^S$, the measurements $\mathbf{y}$, the forward operator $\mathbf{A}$, and possibly other algorithm-internal parameters such as a scalar that weights likelihood and prior terms or parameters that define the noise schedule. This template is summarized in algorithm 2 and specialized instances for the step $\mathcal{S}$ that correspond to the three popular algorithms Chung diffusion posterior sampling (C-DPS) (Chung et al., 2023), diffusion models for plug-and-play image restoration (DiffPIR) (Zhu et al., 2023), and DPnP (Xu & Chi, 2024) are given in appendix E.2. We have absorbed the (variance-preserving) scaling into the step $\mathcal{S}$ since this template is not fundamentally limited to diffusion processes but supports any (also not monotonically decreasing) noise schedules. In addition, noise variances $\{\sigma_t\}_{t=1}^T$ are usually derived from the algorithm-internal parameters $\boldsymbol{\lambda}$ that may include a noise schedule.

Algorithm 2 Template for DPS algorithms.

Require: Initial point $\mathbf{x}_T$, $\mathbf{y}$, $\mathbf{A}$, $\boldsymbol{\lambda}$
1: **for** $t = T, \dots, 1$ **do** ▷ *Diffusion process*
2: Sample $\{\bar{\mathbf{x}}_s\}_{s=1}^S \sim p_{\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}_t}$
3: Update $\mathbf{x}_{t-1} = \mathcal{S}(\mathbf{x}_t, \{\bar{\mathbf{x}}_s\}_{s=1}^S, \mathbf{y}, \mathbf{A}, \boldsymbol{\lambda}, t)$
4: **return** $\hat{\mathbf{x}}^{\mathrm{alg}} = \mathbf{x}_0$ ▷ *Posterior sample*

Through this construction, DPS algorithms can use any statistic R of the samples $\{\bar{\mathbf{x}}_s\}_{s=1}^S$ in their update steps. Most methods use the mean $R(\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_S) = \frac{1}{S} \sum_{s=1}^S \bar{\mathbf{x}}_s := \bar{\boldsymbol{\mu}}$, which is the Monte Carlo estimate of $\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}_t]$. An example of a DPS algorithm that utilizes additional statistics is C-DPS, which requires the Jacobian of $\mathbf{x}_t \mapsto \mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}_t]$. As we show in appendix E.1, this Jacobian equals (up to the known VP scaling) the conditional covariance of $\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}_t$, an unbiased estimator of which can be obtained through the statistic $R(\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_S) = \frac{1}{S-1} \sum_{s=1}^S (\bar{\mathbf{x}}_s - \bar{\boldsymbol{\mu}})(\bar{\mathbf{x}}_s - \bar{\boldsymbol{\mu}})^T$. An example of a DPS algorithm that utilizes an al-

ternative statistic is the DPnP algorithm that alternately samples from $p_{\mathbf{X}_0|\mathbf{X}_t=\mathbf{x}_t}$ and a data-proximal problem. There, $R(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_S) = \bar{\mathbf{x}}_1$ is used to obtain one sample from $p_{\mathbf{X}_0|\mathbf{X}_t=\mathbf{x}_t}$.

4 Numerical Experiments

We consider four inverse problems that are frequently encountered in various estimation tasks throughout the natural sciences: denoising, deconvolution, imputation, and reconstruction from partial Fourier measurements. Details about the operators are provided in appendix F.1 and precise descriptions of the benchmarking pipeline (*e.g.*, the number of training, validation, and test signals, and the number of iterations in the Gibbs methods) in appendix F.2.

4.1 Reconstruction Algorithms

The model-based methods and the DPS algorithms require the tuning of some hyperparameters. These were found by grid search on validation data independently for each algorithm, increment distribution, and forward operator. The precise setup for this grid search is given in appendix F.5. Importantly, the hyperparameters for the DPS algorithms were tuned to the learned denoiser due to resource constraints and we view full oracle-tuning as a community task. Parameters obtained with this procedure are later denoted with a star in the superscript.

Model-Based Methods As baseline reconstruction algorithms we consider the model-based methods

$$\hat{\mathbf{x}}^{\ell_2}(\mathbf{y}, \lambda) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d} \left(\tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \lambda\|\mathbf{D}\mathbf{x}\|^2\right), \tag{15}$$

and

$$\hat{\mathbf{x}}^{\ell_1}(\mathbf{y}, \lambda) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d} \left(\tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \lambda\|\mathbf{D}\mathbf{x}\|_1\right), \tag{16}$$

which coincide with the maximum-a-posteriori (MAP) estimators of Lévy processes associated with Gaussian and Laplace increment distributions, respectively.

Diffusion Posterior Sampling Algorithms We consider three DPS algorithms that are popular in the literature. First, the C-DPS algorithm due to Chung et al. (2023), which was one of the first algorithms that was proposed for the resolution of general noisy inverse problems with diffusion priors. Second, the DiffPIR algorithm due to Zhu et al. (2023) that can be regarded as an extension of the C-DPS algorithm and typically reports superior results in standard perception-based evaluations. Third, the DPnP algorithm due to Xu & Chi (2024) that alternates between sampling the denoising subproblem and a data-proximal subproblem. We include the DPnP algorithm to showcase the broad applicability of our framework to nonstandard setups that utilize various statistics of the denoising posterior.

For each DPS algorithm, we benchmark two variants: One where the denoising posterior is sampled with the gold-standard Gibbs methods ("oracle" denoiser) and statistics are computed from those samples, and one where the sampling (or the direct estimation of any point estimate) is done with learned components. For the former, we ensured that the Monte Carlo error (in the estimation of the denoising-posterior expectation) is significantly below the error of the learned denoisers with a rigorous protocol that is described in appendix F.4. For the latter, learning details are provided in section appendix F.3.

Gold-Standard Gibbs Methods The Gibbs methods are used to obtain gold-standard samples from the posterior. As described in section 3, the Gibbs methods are parameter- and bias-free and efficient and, consequently, well-suited for this purpose. Chain lengths, diagnostics, and implementation details are given in appendix F.2; we reuse the same settings across operators and increment families.

4.2 Results

Before advancing, we introduce some notation. For any given measurement $\mathbf{y}$, any DPS algorithm alg that depends on any parameters $\boldsymbol{\lambda}$ produces samples denoted $\{\hat{\mathbf{x}}_k^{\text{alg}}(\mathbf{y}, \boldsymbol{\lambda})\}_{k=1}^{N_{\text{samples}}}$. We denote

Table 1: MMSE optimality gap in decibel (mean $\pm$ standard deviation; lower is better; 0 is a perfect reconstruction) of various estimation methods over the test set. Bold: best among DPS algorithms.

		Gauss$(0, 0.25)$	Laplace(1)	BL$(0.1, 1)$	St(1)	St(2)	St(3)
Denoising	C-DPS	$\mathbf{0.12 \pm 0.18}$	0.12 ± 0.20	2.22 ± 2.26	3.26 ± 1.01	0.28 ± 0.30	0.10 ± 0.18
	DiffPIR	0.16 ± 0.21	$\mathbf{0.09 \pm 0.16}$	$\mathbf{0.72 \pm 1.10}$	$\mathbf{0.93 \pm 1.06}$	$\mathbf{0.07 \pm 0.14}$	0.15 ± 0.21
	DPnP	0.24 ± 0.25	0.11 ± 0.17	1.33 ± 2.12	1.19 ± 1.38	0.10 ± 0.17	$\mathbf{0.10 \pm 0.17}$
	ℓ_1	0.15 ± 0.21	0.06 ± 0.12	3.44 ± 2.38	0.38 ± 0.43	0.14 ± 0.19	0.11 ± 0.18
	ℓ_2	0.00 ± 0.01	0.16 ± 0.21	8.61 ± 3.10	3.25 ± 0.99	0.74 ± 0.83	0.25 ± 0.33
Deconvolution	C-DPS	0.12 ± 0.20	0.12 ± 0.23	4.30 ± 3.87	18.30 ± 5.28	0.46 ± 1.40	0.17 ± 0.53
	DiffPIR	$\mathbf{0.07 \pm 0.17}$	$\mathbf{0.07 \pm 0.19}$	$\mathbf{1.09 \pm 2.22}$	10.45 ± 6.10	$\mathbf{0.09 \pm 0.57}$	$\mathbf{0.08 \pm 0.26}$
	DPnP	0.10 ± 0.18	0.13 ± 0.22	1.71 ± 2.49	$\mathbf{7.84 \pm 5.66}$	0.35 ± 1.39	0.14 ± 0.41
	ℓ_1	1.65 ± 0.84	1.38 ± 0.86	1.86 ± 3.14	1.87 ± 4.01	1.10 ± 1.19	1.28 ± 0.94
	ℓ_2	0.00 ± 0.01	0.07 ± 0.23	6.11 ± 4.49	21.50 ± 4.46	1.44 ± 2.85	0.36 ± 1.09
Imputation	C-DPS	0.15 ± 0.29	0.18 ± 0.39	2.99 ± 2.82	23.33 ± 8.69	0.50 ± 1.09	0.14 ± 0.57
	DiffPIR	$\mathbf{0.09 \pm 0.23}$	$\mathbf{0.08 \pm 0.24}$	$\mathbf{0.24 \pm 1.14}$	$\mathbf{0.88 \pm 3.50}$	$\mathbf{0.11 \pm 0.62}$	$\mathbf{0.08 \pm 0.42}$
	DPnP	0.14 ± 0.32	0.17 ± 0.36	0.50 ± 1.28	10.89 ± 5.92	0.25 ± 0.82	0.27 ± 0.58
	ℓ_1	1.74 ± 1.12	1.77 ± 1.35	1.25 ± 2.78	13.32 ± 5.32	1.37 ± 2.56	1.55 ± 1.58
	ℓ_2	0.00 ± 0.01	0.01 ± 0.05	1.10 ± 1.88	0.42 ± 0.95	0.06 ± 0.34	0.02 ± 0.28
Fourier	C-DPS	0.15 ± 0.36	0.26 ± 0.65	5.90 ± 4.41	4.29 ± 5.78	0.53 ± 0.83	0.35 ± 0.77
	DiffPIR	$\mathbf{0.11 \pm 0.29}$	$\mathbf{0.08 \pm 0.31}$	$\mathbf{0.83 \pm 1.44}$	3.19 ± 4.37	$\mathbf{0.11 \pm 0.39}$	$\mathbf{0.12 \pm 0.37}$
	DPnP	0.11 ± 0.35	0.20 ± 0.51	1.88 ± 2.47	$\mathbf{2.45 \pm 4.83}$	0.39 ± 0.89	0.24 ± 0.64
	ℓ_1	1.50 ± 1.59	0.73 ± 0.94	3.57 ± 2.82	1.07 ± 2.98	0.71 ± 0.99	0.78 ± 0.97
	ℓ_2	0.00 ± 0.02	0.36 ± 0.73	12.22 ± 4.53	9.47 ± 8.34	2.66 ± 3.57	1.03 ± 1.79

$\hat{\mathbf{x}}^{\mathrm{alg}}_{\mathrm{MMSE}}(\mathbf{y}, \boldsymbol{\lambda}) \coloneqq \frac{1}{N_{\mathrm{samples}}} \sum_{k=1}^{N_{\mathrm{samples}}} \hat{\mathbf{x}}^{\mathrm{alg}}_k(\mathbf{y}, \boldsymbol{\lambda})$. For an estimation method $\hat{\mathbf{x}}^{\mathrm{est}}(\,\cdot\,)$ and data $\mathbf{y}$ with corresponding data-generating signal $\mathbf{x}$ we measure the MMSE optimality gap (in decibel) defined by

$$10 \log_{10} \left(\frac{\|\hat{\mathbf{x}}^{\mathrm{est}}(\mathbf{y}) - \mathbf{x}\|^2}{\|\hat{\mathbf{x}}^{\mathrm{Gibbs}}_{\mathrm{MMSE}}(\mathbf{y}) - \mathbf{x}\|^2} \right), \tag{17}$$

where $\hat{\mathbf{x}}^{\mathrm{est}}(\mathbf{y}) = \hat{\mathbf{x}}^{\ell_{1/2}}(\mathbf{y}, \lambda^\star)$ for model-based methods and $\hat{\mathbf{x}}^{\mathrm{alg}}_{\mathrm{MMSE}}(\mathbf{y}, \boldsymbol{\lambda}^\star)$ for DPS algorithms. A gap of 0 indicates a perfect recovery of the gold-standard MMSE estimate and the positive nonzero values show the orders of magnitude of the error relative to the reference error. We found that $N_{\mathrm{samples}} = 50$ provided a good tradeoff between runtime and accuracy by benchmarking the gold-standard Gibbs method with that number of samples.

We report the mean and standard deviation of the MMSE optimality gap over all signal-measurement pairs $(\mathbf{x}, \mathbf{y})$ in the test set obtained by the model-based methods and the DPS algorithms endowed with the learned denoiser in table 1. The Gaussian increment distribution validates the implementation: Since the MMSE and the MAP point estimates coincide, the model-based ℓ_2 estimator matches the Gibbs reference up to the error due to the finite parameter-grid resolution. When the posterior mean is smooth (*e.g.*, imputation and some deconvolution cases), ℓ_2 is the best model-based choice and frequently outperforms the DPS algorithms. When the posterior mean is close to piecewise-constant (typical in denoising of signals with sparse increments), the ℓ_1 estimator is preferred. Among DPS algorithms, DiffPIR is typically the top performer and often exceeds ℓ_2 and ℓ_1 baselines in deconvolution, imputation, and reconstruction from partial Fourier measurements. For spike-and-slab settings (Bernoulli–Laplace), DPS algorithms substantially outperform the model-based baselines across operators. In deconvolution and reconstruction from partial Fourier measurements, DPS algorithms frequently match or surpass the best model-based estimator.

In addition to the reconstruction performance obtained with the learned denoisers—for which the parameters of the algorithms were tuned—we inspect the robustness of the algorithms when replacing the learned denoiser with the oracle denoiser. Here, we discuss general trends; an exhaustive quantitative evaluation is given in appendix G. DPnP is the most robust to swapping the learned denoiser with the oracle denoiser and significantly benefits from the oracle denoiser in the most challenging cases of the spike-and-slab and the extremely heavy-tailed $\mathrm{St}(1)$ increment distributions. By contrast, C-DPS and DiffPIR can require retuning when the denoiser changes: scores can deteriorate after replacing the learned denoiser with the oracle denoiser, whereas a brief hand-tuning of the hyperparameters on the validation set improves them way beyond the learned denoiser (*e.g.*, for DiffPIR and $\mathrm{St}(1)$ increments, the hand-tuning decreased the optimality gap by almost $10\,\mathrm{dB}$). The differences between the algorithms are generally greater than the differences between the learned and oracle variants except for the heavy-tailed cases, which confirms the findings in (Bohra et al., 2023) and indicates that the research of efficient and robust DPS algorithms is still crucial. Qualitative examples of the MMSE estimates and the marginal variances obtained by the DPS algorithms and the gold-standard Gibbs methods are shown in figs. 8 to 15 in the appendix.

Prototypical samples and the corresponding MMSE estimate obtained from a DPS algorithm (here DiffPIR for deconvolution of a signal with $\mathrm{BL}(0.1, 1)$ increments) are shown in fig. 3. The full conditional reverse-diffusion trajectory, the data-generating signal, the measurements, and the MMSE estimated obtained with the gold-standard Gibbs methods are shown in fig. 16 in the appendix. The figure highlights a key distinction: Posterior *samples* often preserve high-frequency structure and reflect prior variability, whereas the *MMSE point estimate*—obtained by averaging all samples—is much smoother. This explains why DPS methods tend to score higher on perception-oriented metrics, while regressors that target the MMSE point estimate (through training with the mean squared error) excel on distortion metrics like the peak signal-to-noise-ratio (PSNR)). Consistent with this distinction, Saharia et al. (2023) fairly compare a sampling-based method to an MMSE regressor and find the expected trade-off: higher PSNR and structural similarity for the regressor and better perceptual scores for the sampler. We therefore recommend making the Bayesian target explicit—point estimate versus sample quality—and using evaluation protocols that are aligned to that target. Our framework supports this by offering gold-standard posterior samples and oracle denoisers.

In addition to the evaluation of the MMSE optimality gap, which is on the point-estimator level, we analyze the highest-posterior-density coverage of the algorithms. Specifically, for any measurement $\mathbf{y}$ and any $k = 1, \dots, N_{\text{samples}}$, denote $l_k(\mathbf{y}) \coloneqq \log p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\hat{\mathbf{x}}^{\text{alg}}_{P(k)}(\mathbf{y}, \boldsymbol{\lambda}^{\text{alg},\star}))$[3] where P is the permutation that ensures that $l_1(\mathbf{y}) \geq l_2(\mathbf{y}) \geq \cdots \geq l_{N_{\text{samples}}}(\mathbf{y})$ and define the empirical highest-posterior-density threshold at $\alpha \in [0, 1]$ as $l_{\lceil \alpha N_{\text{samples}} \rceil}(\mathbf{y})$. We declare the data-generating signal $\mathbf{x}$ covered if $\log p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x}) \geq l_{\lceil \alpha N_{\text{samples}} \rceil}(\mathbf{y})$ and define the coverage of a method as the fraction of signal-measurement pairs $(\mathbf{x}, \mathbf{y})$ in the test set for which $\mathbf{x}$ is covered by the threshold $l_{\lceil \alpha N_{\text{samples}} \rceil}(\mathbf{y})$. The coverage of a calibrated posterior-sampling method will be α up to Monte Carlo error. A coverage result that is significantly less than α indicates that the samples obtained by the method concentrate too heavily around the mode; a coverage result that is greater than α indicates that the samples are too spread out. We again discuss general trends here and present an exhaustive quantitative evaluation in appendix G. The coverages obtained by the DPS algorithms are generally much smaller than α, which indicates that they are uncalibrated and is in line with what is reported by Thong et al. (2024). For C-DPS and DiffPIR, the reported coverage values are almost always 0 except for $\mathrm{BL}(0.1, 1)$ and $\mathrm{St}(1)$ increments, where the coverages are usually (close to) 1 for C-DPS and inconsistent for DiffPIR. For almost all increment distributions and forward operators, DPnP reports coverage values that are closest to but typically smaller than α.

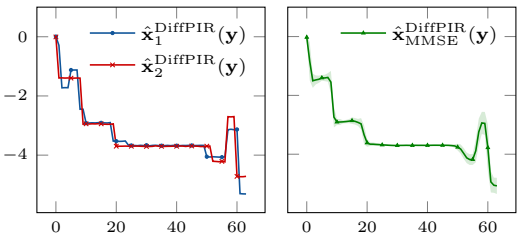

Figure 3: Conditional generation for deconvolution of a signal with $\mathrm{BL}(0.1, 1)$ increments with DiffPIR. The shaded area indicates the variance.

5 Conclusion

We introduced a statistical benchmark for diffusion posterior sampling algorithms for linear inverse problems. The framework proceeds by constructing signals with a known distribution, simulating the measurement process, and subsequently generating samples from the posterior distribution that arises through the combination of the known prior and the known likelihood. Gold-standard samples from this distribution are obtained via efficient Gibbs methods, and these samples are then compared to those obtained by the diffusion posterior sampling algorithms. In addition, the Gibbs methods can serve as oracle MMSE denoisers within the denoising posteriors encountered in each iteration of the reverse SDE. Consequently, the framework also enables the isolation and quantification of the error attributable to the likelihood approximations in the conditional reverse diffusion. We provided numerical results for three common diffusion posterior sampling algorithms applied to four common inverse problems. A consistent theme across all tested algorithms is that they are not calibrated, which demonstrates that research into algorithms that perform better in this respect remains crucial. We invite other researchers to benchmark their algorithms on our open implementation, which is deliberately designed so that novel DPS algorithms can be evaluated in a plug-and-play manner.

[3] With slight abuse of notation, $\log p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$ is the unnormalized ground-truth log-posterior (10); the additive constant is the same across all methods so ranking is valid.

Reproducibility Statement We release an online repository with complete algorithm implementations and step-by-step instructions to reproduce all results. A containerized runtime enables one-command setup and fully automated execution via the provided scripts. Each algorithm is specified at a level that supports independent reimplementation: the main text precisely details GLM sampling, and the appendix presents the Bernoulli–Laplace Gibbs method using implementation-aligned notation, together with practical optimizations required for acceptable runtimes. The appendix also enumerates all experimental settings, including the numbers of training/validation/test signals, the samples-per-datum for each sampler, and the exact grid-search procedure used to select hyperparameters.

Usage of Large Language Models We used large language models to adapt passages of already-written text for readability and conciseness.

REFERENCES

Amira Alakhdar, Barnabas Poczos, and Newell Washburn. Diffusion models in de novo drug design. *Journal of Chemical Information and Modeling*, 64(19):7238–7256, September 2024. ISSN 1549-960X. doi: 10.1021/acs.jcim.4c01107. URL http://dx.doi.org/10.1021/acs.jcim.4c01107.

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: 10.1016/0304-4149(82)90051-5.

Robert Bassett and Julio Deride. Maximum a posteriori estimators as a limit of Bayes estimators. *Mathematical Programming*, 174(1–2):129–144, January 2018. ISSN 1436-4646. doi: 10.1007/s10107-018-1241-0. URL http://dx.doi.org/10.1007/s10107-018-1241-0.

Robert C. Blattberg and Nicholas J. Gonedes. A comparison of the stable and student distributions as statistical models for stock prices. *The Journal of Business*, 47(2):244–280, 1974. ISSN 00219398, 15375374. URL http://www.jstor.org/stable/2353383.

Lea Bogensperger, Dominik Narnhofer, Ahmed Allam, Konrad Schindler, and Michael Krauthammer. A variational perspective on generative protein fitness optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=fINjgBMnTS.

Pakshal Bohra, Pol del Aguila Pla, Jean-François Giovannelli, and Michael Unser. A statistical framework to investigate the optimality of signal-reconstruction methods. *IEEE Transactions on Signal Processing*, 71:2043–2055, 2023. doi: 10.1109/TSP.2023.3282062.

George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, August 1992. ISSN 1537-2731. doi: 10.1080/00031305.1992.10475878. URL http://dx.doi.org/10.1080/00031305.1992.10475878.

Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated MRI. *Medical Image Analysis*, 80:102479, 2022. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102479. URL https://www.sciencedirect.com/science/article/pii/S1361841522001268.

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.

Christian Clason, Tapio Helin, Remo Kretschmann, and Petteri Piiroinen. Generalized modes in bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2):652–684, 2019. doi: 10.1137/18M1191804. URL https://doi.org/10.1137/18M1191804.

R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001. doi: 10.1080/713665670. URL https://doi.org/10.1080/713665670.

Evan Scope Crafts and Umberto Villa. Benchmarking diffusion annealing-based Bayesian inverse problem solvers. *IEEE Open Journal of Signal Processing*, 6:975–991, 2025. doi: 10.1109/OJSP.2025.3597867.

Giannis Daras, Alexandros G. Dimakis, and Constantinos Daskalakis. Consistent diffusion meets Tweedie: Training exact ambient diffusion models with noisy data. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Luc Devroye. Random variate generation for the generalized inverse Gaussian distribution. *Statistics and Computing*, 24(2):239–246, December 2012. ISSN 1573-1375. doi: 10.1007/s11222-012-9367-z. URL http://dx.doi.org/10.1007/s11222-012-9367-z.

Julius Erbach, Dominik Narnhofer, Andreas Dombos, Bernt Schiele, Jan Eric Lenssen, and Konrad Schindler. Solving inverse problems with FLAIR, 2025. URL https://arxiv.org/abs/2506.02680. arXiv.

Xuefeng Gao, Hoang M. Nguyen, and Lingjiong Zhu. Wasserstein convergence guarantees for a general class of score-based generative models. *Journal of Machine Learning Research*, 26(43):1–54, 2025. URL http://jmlr.org/papers/v26/24-0902.html.

Crispin W Gardiner. *Handbook of stochastic methods: For physics, chemistry and the natural sciences*. Springer, Berlin, Germany, 1990.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, November 2013. ISBN 9780429113079. doi: 10.1201/b16018. URL http://dx.doi.org/10.1201/b16018.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Desmond J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001. doi: 10.1137/S0036144500378302. URL https://doi.org/10.1137/S0036144500378302.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Hongbin Huang, Minghua Chen, and Xiao Qiao. Generative learning for financial time series with irregular and scale-invariant patterns. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CdjnzWsQax.

Samuel Hurault, Matthieu Terris, Thomas Moreau, and Gabriel Peyré. From score matching to diffusion: A fine-grained error analysis in the Gaussian setting, 2025. URL https://arxiv.org/abs/2503.11615. arXiv.

Achim Klenke. *Probability Theory: A Comprehensive Course*. Springer International Publishing, 2020. ISBN 9783030564025. doi: 10.1007/978-3-030-56402-5. URL http://dx.doi.org/10.1007/978-3-030-56402-5.

Muhamed Kuric, Martin Zach, Andreas Habring, Michael Unser, and Thomas Pock. The Gaussian latent machine: Efficient prior and posterior sampling for inverse problems, 2025. URL https://arxiv.org/abs/2505.12836. arXiv.

Hoang Trieu Vy Le, Marion Foare, Audrey Repetti, and Nelly Pustelnik. Embedding Blake–Zisserman regularization in unfolded proximal neural networks for enhanced edge detection. *IEEE Signal Processing Letters*, 32:1271–1275, 2025. doi: 10.1109/LSP.2025.3547671.

Jean-Marie Lemercier, Julius Richter, Simon Welker, Eloi Moliner, Vesa Välimäki, and Timo Gerkmann. Diffusion models for audio restoration: A review. *IEEE Signal Processing Magazine*, 41(6):72–84, 2024. doi: 10.1109/MSP.2024.3445871.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. 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=3s9IrEsjLyk.

Jiaming Liu, Rushil Anirudh, Jayaraman J. Thiagarajan, Stewart He, K. Aditya Mohan, Ulugbek S. Kamilov, and Hyojin Kim. DOLCE: A model-based probabilistic diffusion framework for limited-angle CT reconstruction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10464–10474, 2023. doi: 10.1109/ICCV51070.2023.00963.

Ozan Özdenizci and Robert Legenstein. Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–12, 2023. doi: 10.1109/TPAMI.2023.3238179.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Emile Pierret and Bruno Galerne. Exact evaluation of the accuracy of diffusion models for inverse problems with Gaussian data distributions, 2025a. URL https://arxiv.org/abs/2507.07008. arXiv.

Emile Pierret and Bruno Galerne. Diffusion models for Gaussian distributions: Exact solutions and Wasserstein errors. In *Forty-second International Conference on Machine Learning*, 2025b.

R. Po, W. Yifan, V. Golyanik, K. Aberman, J. T. Barron, A. Bermano, E. Chan, T. Dekel, A. Holynski, A. Kanazawa, C.K. Liu, L. Liu, B. Mildenhall, M. Nießner, B. Ommer, C. Theobalt, P. Wonka, and G. Wetzstein. State of the art on diffusion models for visual computing. *Computer Graphics Forum*, 43(2), April 2024. ISSN 1467-8659. doi: 10.1111/cgf.15063. URL http://dx.doi.org/10.1111/cgf.15063.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on Machine Learning*, pp. 8857–8868. PMLR, 2021.

Mengwei Ren, Mauricio Delbracio, Hossein Talebi, Guido Gerig, and Peyman Milanfar. Multiscale structure guided diffusion for image deblurring. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10687–10699, Los Alamitos, CA, USA, October 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.00984. URL https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00984.

Severi Rissanen, Markus Heinonen, and Arno Solin. Free hunch: Denoiser covariance estimation for diffusion models without extra costs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4JK2XMGUc8.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685. IEEE, June 2022. doi: 10.1109/cvpr52688.2022.01042. URL http://dx.doi.org/10.1109/CVPR52688.2022.01042.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023. doi: 10.1109/TPAMI.2022.3204461.

Ken-Iti Sato. *Lévy processes and infinitely divisible distributions*. Cambridge studies in advanced mathematics. Cambridge University Press, Cambridge, England, November 1999.

Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, and Frank Lenzen. *Variational Methods in Imaging*. Applied mathematical sciences. Springer, New York, NY, 2009 edition, October 2008.

Wim Schoutens. *Lévy processes in finance*. Wiley Series in Probability and Statistics. John Wiley & Sons, Chichester, England, March 2003.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 7 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11895–11907, 2019.

Yang Song, Jasha 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.

Stanislas Strasman, Antonio Ocello, Claire Boyer, Sylvain Le Corff, and Vincent Lemaire. An analysis of the noise schedule for score-based generative models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=BlYIPa0Fx1.

A. M. Stuart. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 2010. doi: 10.1017/S0962492910000061.

David Y. W. Thong, Charlesquin Kemajou Mbakam, and Marcelo Pereyra. Do Bayesian imaging methods report trustworthy probabilities?, 2024. URL https://arxiv.org/abs/2405.08179. arXiv.

Michael Unser and Pouya D. Tafti. *An Introduction to Sparse Stochastic Processes*. Cambridge University Press, 2014.

David A van Dyk and Taeyoung Park. Partially collapsed gibbs samplers. *Journal of the American Statistical Association*, 103(482):790–796, 2008. doi: 10.1198/016214508000000409. URL https://doi.org/10.1198/016214508000000409.

Martin J Wainwright and Eero Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/6a5dfac4be1502501489fc0f5a24b667-Paper.pdf.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Zhuohan Wang and Carmine Ventre. A financial time series denoiser based on diffusion models. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, pp. 72–80, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400710810. doi: 10.1145/3677052.3698649. URL https://doi.org/10.1145/3677052.3698649.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620 (7976):1089–1100, July 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8. URL http://dx.doi.org/10.1038/s41586-023-06415-8.

Xingyu Xu and Yuejie Chi. Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=SLnsoaY4u1.

Duoduo Xue, Wenrui Dai, Ziyang Zheng, Xinyu Peng, Junni Zou, and Hongkai Xiong. Fourier phase retrieval with diffusion priors guided by failure detection. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2025. doi: 10.1109/ISCAS56072.2025.11042933.

Nebiyou Yismaw, Ulugbek S. Kamilov, and M. Salman Asif. Covariance-corrected diffusion models for solving inverse problems. In *2025 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 26–30, 2025. doi: 10.1109/SSP64130.2025.11073300.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (NTIRE)*, 2023.

A Bayes Estimators

A benefit of the Bayesian approach over classical variational methods (see, e.g., (Scherzer et al., 2008)) is that different point estimates arise from a fixed prior. For a given measurement $\mathbf{y}$, these point estimates summarize the posterior distribution $p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$ with respect to a given loss $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ via the optimization problem of finding the point $\hat{\mathbf{x}}_\ell(\mathbf{y})$ that minimizes the posterior risk:

$$\hat{\mathbf{x}}_\ell(\mathbf{y}) = \operatorname*{arg\,min}_{\hat{\mathbf{x}}\in\mathbb{R}^d} \left(\int_{\mathbb{R}^d} \ell(\hat{\mathbf{x}}, \mathbf{x})\, p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x})\, \mathrm{d}\mathbf{x} \right). \tag{18}$$

In this paper, the Bayes estimator with respect to the mean-squared error (MSE) $\ell = \frac{1}{d}\|\cdot - \cdot\|^2$ plays a key role due to its close relation to the prior *score* in the reverse diffusion (see section 2) and because we quantify the performance of DPS algorithms via the MMSE optimality gap in section 4. With this choice of ℓ, (18) can be written as

$$\hat{\mathbf{x}}_{\mathrm{MMSE}}(\mathbf{y}) = \operatorname*{arg\,min}_{\hat{\mathbf{x}}\in\mathbb{R}^d} \left(\int_{\mathbb{R}^d} \tfrac{1}{d}\|\hat{\mathbf{x}}-\mathbf{x}\|^2\, p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x})\, \mathrm{d}\mathbf{x} \right) = \int_{\mathbb{R}^d} \mathbf{x} p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathbb{E}[\mathbf{X} \mid \mathbf{Y} = \mathbf{y}], \tag{19}$$

which is the expectation of the posterior $p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$.

Another widely-used estimator arises through the choice

$$\ell(\hat{\mathbf{x}}, \mathbf{x}) = -\chi_{\{\hat{\mathbf{x}}\}}(\mathbf{x}) \tag{20}$$

where

$$\chi_A(\mathbf{x}) \coloneqq \begin{cases} 1 & \text{if } \mathbf{x} \in A, \\ 0 & \text{else}, \end{cases} \tag{21}$$

which leads to the MAP estimator that seeks the mode of the posterior:[4]

$$\hat{\mathbf{x}}_{\mathrm{MAP}}(\mathbf{y}) = \operatorname*{arg\,min}_{\hat{\mathbf{x}}\in\mathbb{R}^d} \left(\int_{\mathbb{R}^d} -\chi_{\{\hat{\mathbf{x}}\}}(\mathbf{x})\, p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\mathbf{x})\, \mathrm{d}\mathbf{x} \right) = \operatorname*{arg\,max}_{\hat{\mathbf{x}}\in\mathbb{R}^d} p_{\mathbf{X}|\mathbf{Y}=\mathbf{y}}(\hat{\mathbf{x}}). \tag{22}$$

Rewriting (22) as

$$\hat{\mathbf{x}}_{\mathrm{MAP}}(\mathbf{y}) = \operatorname*{arg\,min}_{\hat{\mathbf{x}}\in\mathbb{R}^d} \left(-\tfrac{1}{2\sigma_{\mathrm{n}}^2}\|\mathbf{A}\hat{\mathbf{x}} - \mathbf{y}\|^2 - \log p_{\mathbf{X}}(\hat{\mathbf{x}}) \right), \tag{23}$$

reveals a close relation to classical variational approaches after identifying the regularizer with $-\log p_{\mathbf{X}}$.

[4] This definition is informal but sufficient for the purposes of this paper. For continuous posteriors, the strict 0–1 loss Bayes' rule is ill-posed. A common formalization defines MAP as the limit of Bayes estimators under shrinking small-ball 0–1 losses; under additional regularity, this limit agrees with the posterior mode (Bassett & Deride, 2018; Clason et al., 2019). The MAP estimator may also not be unique.

B TWEEDIE FORMULA

In the setting of section 2, we now derive an equality that relates $\nabla \log p_{\mathbf{X}_t}$ to $\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \cdot\,]$, *i.e.*, the MMSE estimate of $\mathbf{X}_0$ given that $\mathbf{X}_t$ takes on a certain value. Similar derivations can be found in, *e.g.*, (Song et al., 2021; Chung et al., 2023; Daras et al., 2024), but we include it to underscore the relevance of the MMSE estimate in this paper and to facilitate the understanding of its relation to various objects. Under the variance-preserving choice for $\mathbf{f}(\mathbf{x}, t) = -\frac{\beta(t)}{2}\mathbf{x}$ and $g(t) = \sqrt{\beta(t)}$ of the drift and diffusion coefficient, the diffusion SDE (4) simplifies to a time-inhomogeneous Ornstein–Uhlenbeck SDE (see Klenke (2020, Example 26.5))

$$\mathrm{d}\mathbf{X}_t = -\tfrac{\beta(t)}{2}\mathbf{X}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t, \tag{24}$$

whose pathwise solution

$$\mathbf{X}_t = \alpha(t)\mathbf{X}_0 + \int_0^t \frac{\alpha(t)}{\alpha(s)}\sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_s, \tag{25}$$

where $\mathbf{X}_0$ is an appropriate initial condition and $\alpha(t) = \exp\bigl(-\frac{1}{2}\int_0^t \beta(s)\,\mathrm{d}s\bigr)$, can be computed with standard techniques, see, *e.g.*, (Gardiner, 1990, Section 4.4.4). In addition, since

$$\int_0^t \left(\frac{\alpha(t)}{\alpha(s)}\right)^2 \beta(t)\,\mathrm{d}s = \int_0^t \beta(s)\exp\Bigl(-\int_s^t \beta(u)\,\mathrm{d}u\Bigr)\,\mathrm{d}s = 1 - \alpha^2(t) \tag{26}$$

we can write that

$$\mathbf{X}_t = \alpha(t)\mathbf{X}_0 + \sigma(t)\mathbf{N} \tag{27}$$

in distribution, where $\sigma^2(t) = 1 - \alpha^2(t)$. Consequently, the density of $\mathbf{X}_t$ is given by the convolution of $p_{\mathbf{X}_0}$ with a Gaussian with variance $\sigma^2(t)$ and appropriate scaling by $\alpha(t)$:

$$p_{\mathbf{X}_t}(\mathbf{x}) = \int_{\mathbb{R}^d} g_{\mathbf{0},\sigma(t)^2\mathbf{I}}(\mathbf{x} - \alpha(t)\hat{\mathbf{x}})\,p_{\mathbf{X}_0}(\hat{\mathbf{x}})\,\mathrm{d}\hat{\mathbf{x}}, \tag{28}$$

where $g_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) = (2\pi)^{-\frac{d}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\bigl(-\frac{1}{2}\|\mathbf{x}-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}}\bigr)$. Finally, after taking the gradient we see that

$$\begin{aligned}
\nabla p_{\mathbf{X}_t}(\mathbf{x}) &= \int_{\mathbb{R}^d} \nabla g_{\mathbf{0},\sigma(t)^2\mathbf{I}}(\mathbf{x} - \alpha(t)\hat{\mathbf{x}})\,p_{\mathbf{X}_0}(\hat{\mathbf{x}})\,\mathrm{d}\hat{\mathbf{x}} \\
&= \int_{\mathbb{R}^d} \bigl(-\tfrac{1}{\sigma^2(t)}(\mathbf{x} - \alpha(t)\hat{\mathbf{x}})\,g_{\mathbf{0},\sigma^2(t)\mathbf{I}}(\mathbf{x} - \alpha(t)\hat{\mathbf{x}})\,p_{\mathbf{X}_0}(\hat{\mathbf{x}})\,\mathrm{d}\hat{\mathbf{x}} \\
&= -\tfrac{1}{\sigma^2(t)}\Bigl(\mathbf{x}p_{\mathbf{X}_t}(\mathbf{x}) - \alpha(t)\int_{\mathbb{R}^d} \hat{\mathbf{x}}\,g_{\mathbf{0},\sigma^2(t)\mathbf{I}}(\mathbf{x} - \alpha(t)\hat{\mathbf{x}})\,p_{\mathbf{X}_0}(\hat{\mathbf{x}})\,\mathrm{d}\hat{\mathbf{x}}\Bigr) \\
&= -\tfrac{1}{\sigma(t)^2}\bigl(\mathbf{x}p_{\mathbf{X}_t}(\mathbf{x}) - \alpha(t)p_{\mathbf{X}_t}(\mathbf{x})\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]\bigr)
\end{aligned} \tag{29}$$

such that, after dividing by $p_{\mathbf{X}_t}(\mathbf{x})$ and since $\frac{\nabla p_{\mathbf{X}_t}(\mathbf{x})}{p_{\mathbf{X}_t}(\mathbf{x})} = \nabla \log p_{\mathbf{X}_t}(\mathbf{x})$, we find the celebrated Tweedie identity

$$\nabla \log p_{\mathbf{X}_t}(\mathbf{x}) = -\sigma(t)^{-2}\bigl(\mathbf{x} - \alpha(t)\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]\bigr). \tag{30}$$

B.1 A CONNECTION BETWEEN THE DISCRETIZED REVERSE SDE AND DDPM

To show the connection between the Euler–Maruyama discretization of the reverse-diffusion SDE and the DDPM backward process, we start by deriving the latter from the respective forward process. DDPM was been introduced by Sohl-Dickstein et al. (2015) as a discrete-time Markov chain of length T with Gaussian transitions:

$$p_{\mathbf{X}_t|\mathbf{X}_{t-1}=\mathbf{x}_{t-1}} = \mathrm{Gauss}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{31}$$

such that the transitions from $\mathbf{X}_0$ to $\mathbf{X}_t$ are also tractable as

$$\mathbf{X}_t = \sqrt{\bar{\alpha}_t}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{Z}_t, \tag{32}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=0}^t \alpha_s$ and $\mathbf{Z}_t \sim \mathrm{Gauss}(\mathbf{0}, \mathbf{I})$. By definition,

$$\mathbf{X}_t = \sqrt{1-\beta_t}\mathbf{X}_{t-1} + \sqrt{\beta_t}\mathbf{Z}_{t-1} \tag{33}$$

and a straightforward application of Tweedie's formula (6) gives that

$$\mathbb{E}[\mathbf{X}_{t-1}|\mathbf{X}_t] = \tfrac{1}{\sqrt{\alpha_t}}\big(\mathbf{X}_t + (1-\alpha_t)\nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t)\big), \tag{34}$$

which leads to the DDPM backward transitions

$$\mathbf{X}_{t-1} = \tfrac{1}{\sqrt{1-\beta_t}}\left(\mathbf{X}_t + \beta_t \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t)\right) + \sqrt{\beta_t}\mathbf{Z}_t \tag{35}$$

like they appear in (7) in the main text.

Now, we recall the reverse-diffusion SDE, which under our choice of the drift and diffusion coefficient is given by

$$\mathrm{d}\mathbf{X}_t = \big(-\tfrac{\beta(t)}{2}\mathbf{X}_t - \beta(t)\nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t)\big)\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t. \tag{36}$$

A first-order step from t to $t-1$ ($\mathrm{d}t = -1$) of gives the Euler–Maruyama update

$$\mathbf{X}_{t-1} = \big(1 + \tfrac{\beta_t}{2}\big)\mathbf{X}_t + \beta_t \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t) + \sqrt{\beta_t}\mathbf{Z}_t, \tag{37}$$

where $\beta_t \coloneqq \beta(t)$ and $\mathbf{Z}_t \sim \mathrm{Gauss}(\mathbf{0}, \mathbf{I})$.

The DDPM reverse process (35) can be related to the the Euler–Maruyama discretization of the reverse SDE (37) via Taylor expansions since

$$\frac{1}{\sqrt{1-\beta_t}} = 1 + \frac{\beta_t}{2} + \mathcal{O}(\beta_t^2) \tag{38}$$

and

$$\frac{\beta_t}{\sqrt{1-\beta_t}} = \beta_t + \mathcal{O}(\beta_t^2) \tag{39}$$

as $\beta_t \to 0$.

C Lévy Processes and Increment Distributions

The prior distributions in our framework are those of signals obtained by regularly spaced samples of processes with independent, stationary increments (Lévy processes and their discrete-time counterparts). We briefly recall the definition; see Unser & Tafti (2014); Sato (1999) for background and the link to infinitely divisible laws.

Definition C.1 (Lévy process)**.** A stochastic process $s = \{s(t) : t \geq 0\}$ is a Lévy process if

1. $s(0) = 0$ almost surely;
2. (independent increments) for any $N \in \mathbb{N} \setminus \{0, 1\}$ and $0 \leq t_1 < t_2 < \cdots < t_N < \infty$, the increments $(s(t_2) - s(t_1)), (s(t_3) - s(t_2)), \ldots, (s(t_N) - s(t_{N-1}))$ are mutually independent;
3. (stationary increments) for any given step h, the increment process $u_h = \{s(t) - s(t-h) : t > h\}$ is stationary;
4. (stochastic continuity) for any $\varepsilon > 0$ and $t \geq 0$

$$\lim_{h \to 0} \Pr\big(|s(t+h) - s(t)| > \varepsilon\big) = 0.$$

We form discrete signals by sampling s at integer times and stacking the values into $\mathbf{x} = (s(1), \ldots, s(d))$. Let the unit-step increments be $\mathbf{u}_k = s(k) - s(k-1)$ for $k = 1, \ldots, d$. By independence and stationarity, the law of $\mathbf{u}_k$ does not depend on k and we denote it p_U.[5] We define the finite-difference matrix

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \tag{40}$$

[5] For our choices, it always has a density w.r.t. a suitable reference measure.

Table 2: Summary of univariate distributions used throughout this work. Parameters appear in the order they are specified in this table, *e.g.* $\mathrm{Gauss}(\mu, \sigma^2)$.

Name	Distribution	Parameter(s)	Supp.	Notation
Gaussian	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right)$	$\mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}_{>0}$	$\mathbb{R}$	Gauss
Exponential	$\lambda \exp(-\lambda x)$	$\lambda \in \mathbb{R}_{>0}$	$\mathbb{R}_{\geq 0}$	Exp
Laplace	$\frac{1}{2b} \exp\left(-\frac{\lvert x\rvert}{b}\right)$	$b \in \mathbb{R}_{>0}$	$\mathbb{R}$	Laplace
Student-t	$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	$\nu \in \mathbb{R}_{>0}$	$\mathbb{R}$	St
Gamma	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x)$	$\alpha, \beta \in \mathbb{R}_{>0}$	$\mathbb{R}_{>0}$	Gamma
Gen. inv. Gaussian	$\frac{(\frac{a}{b})^{\frac{p}{2}}}{2K_p(\sqrt{ab})} x^{p-1} \exp\left(-\frac{ax+b/x}{2}\right)$	$a, b \in \mathbb{R}_{>0}, p \in \mathbb{R}$	$\mathbb{R}_{>0}$	GIG
Bernoulli–Laplace	$\lambda\delta(x) + (1-\lambda)\frac{1}{2b}\exp(-\frac{\lvert x\rvert}{b})$	$\lambda \in [0,1], b \in \mathbb{R}_{>0}$	$\mathbb{R}$	BL

Γ denotes the gamma function defined as $\Gamma(x) = \int_0^\infty t^{x-1}\exp(-t)\,\mathrm{d}x$ for any $x \in \mathbb{R}_{>0}$.
K_ν denotes the modified Bessel function of the second kind with parameter ν.

such that the increment vector satisfies

$$\mathbf{u} = \mathbf{D}\mathbf{x}. \tag{41}$$

Because $s(0) = 0$, the finite-difference matrix $\mathbf{D}$ has an initial condition which makes it invertible and $\mathbf{D}^{-1}$ is a lower-triangular matrix of ones, which also implies that for all $k = 1, 2, \ldots, d$,

$$\mathbf{x}_k = \sum_{n=1}^{k} \mathbf{u}_n, \tag{42}$$

which is a convenient way to synthesize signals once $\mathbf{u}$ is drawn. The combination of (41) with the independence of the increments implies that the density of the discrete signal is

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{k=1}^{d} p_U\big((\mathbf{D}\mathbf{x})_k\big). \tag{43}$$

C.1 Extensions

The approach in this paper can be extended to two- or higher-dimensional signals on grids, such as images or videos, and even more specialized structures like signals defined over trees or more general graphs. The structure of the signal is effectively encoded through the choice of the matrix $\mathbf{D}$. For instance, a two-dimensional finite-difference matrix would result in a signal vector that can be interpreted as a two-dimensional image. The main additional (computational) challenge is sampling during signal generation: Whenever $\mathbf{D}$ is not trivially reducible to a one-dimensional operator, the resulting model (43) will be overcomplete and, in general, no whitening transformation exists to decouple increments for independent sampling. The extension to higher-dimensional signals and the complications that arise in that context are rigorously treated in Kuric et al. (2025).

C.2 Latent Distributions and Notation

Some of the distributions that we rely on in this work have multiple competing parametrizations. To avoid ambiguities, we provide precise definitions of the four increment distributions that we consider in this work—Gaussian, Laplace, Student-t, and Bernoulli–Laplace (spike-and-slab)—and our notation of these and other distributions that we use in this work in table 2. Table 3 lists the latent maps and conditional latent distributions that are needed for the GLM for the distributions in this work.

D Gibbs Methods and Sampling Efficiency

Gibbs methods are Markov chain Monte Carlo (MCMC) methods to sample from a joint distribution $p_{\mathbf{X},\mathbf{Z}_1,\mathbf{Z}_2,\ldots,\mathbf{Z}_n}$ of $(n+1)$ variables that are advantageous when the direct sampling is computationally

Table 3: Latent variable representations and conditional distributions for common distributions.

Dist. ϕ_i	Latent dist. f_i	Latent maps	Cond. latent dist. $p_{\mathbf{Z}_i \mid X=(\mathbf{K}\mathbf{x})_i}$
$\mathrm{Gauss}(\mu, \sigma^2)$	$\delta(0)$	$\mu_i(z_i) = \mu,\ \sigma_i^2(z) = \sigma^2$	$\delta(0)$
$\mathrm{Laplace}(b)$	$\mathrm{Exp}\left(\frac{1}{2b^2}\right)$	$\mu_i(z_i) = 0,\ \sigma_i^2(z_i) = z_i$	$\mathrm{GIG}\left(\frac{1}{b^2}, (\mathbf{K}\mathbf{x})_i^2, \frac{1}{2}\right)$
$\mathrm{St}(\nu)$	$\mathrm{Gamma}\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$	$\mu_i(z_i) = 0,\ \sigma_i^2(z_i) = \frac{1}{z_i}$	$\mathrm{Gamma}\left(\frac{\nu+1}{2}, \frac{\nu+(\mathbf{K}\mathbf{x})_i^2}{2}\right)$

Algorithm 3 Latent-variable Gibbs sampling of $p_{\mathbf{X},\mathbf{Z}_1,\ldots,\mathbf{Z}_N}$.

Require: Burn-in period $B \in \mathbb{N}$, number of samples $S \in \mathbb{N}$, initial point $(\mathbf{x}_0, \mathbf{z}_1, \ldots\ \mathbf{z}_N)$.

1: **for** $s = 1, 2, \ldots, B + S$ **do**
2: $\quad \mathbf{x}_s \sim p_{\mathbf{X} \mid \mathbf{Z}_1 = \mathbf{z}_1, \ldots, \mathbf{Z}_N = \mathbf{z}_N}$
3: $\quad \mathbf{z}_1 \sim p_{\mathbf{Z}_1 \mid \mathbf{X} = \mathbf{x}_k, \ldots, \mathbf{Z}_N = \mathbf{z}_N}$
4: $\quad \vdots$
5: **return** $\{\mathbf{x}_{B+s}\}_{s=1}^{S}$

difficult but sampling from the conditional distributions $p_{\mathbf{X} \mid \mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n}, p_{\mathbf{Z}_1 \mid \mathbf{X}, \mathbf{Z}_2, \ldots, \mathbf{Z}_n}, \ldots$ is easy. Gibbs methods cycle through the conditional distributions with repeated draws, which maintains the joint distribution invariant (Casella & George, 1992). The naming of the variables $\mathbf{X}, \mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n$ is deliberately chosen to emphasize that we use *latent-variable* Gibbs methods that rely on auxiliary variables that are introduced solely to make the conditionals simple. The steps of a general latent variable Gibbs sampler are shown in algorithm 3, where the iteration counter in the sampling of the latent variables is omitted since they need not be stored and previous iterations can immediately be overwritten.

Kuric et al. (2025) recently showed that such methods are significantly faster than other standard sampling routines that are commonly used in settings similar to the one in this paper. They report sampling efficiencies of close to 1, while alternatives, such as the Metropolis-adjusted Langevin algorithm, achieve sampling efficiencies of around 1×10^{-3}.[6] In addition, Gibbs methods require no step-size or acceptance-rate tuning and introduce no discretization bias. These properties motivate our use of Gibbs methods for the fast and robust posterior sampling throughout this work.

Like all MCMC methods, in practice Gibbs methods benefit from discarding some number of initial samples, the *burn-in period*, when the initial point is located in low-density regions. After the burn-in period, it is crucial to tune the number of samples such that the Monte Carlo estimates of various quantities, such as the MMSE estimate in (19), are sufficiently accurate. We discuss our choice of the burn-in period and the number of samples for the various problems in appendix F.2.

D.1 A Gibbs Method for Bernoulli–Laplace Increments

We start by noting that the Bernoulli–Laplace density

$$p_U(u) = \lambda \delta(u) + (1 - \lambda) \tfrac{b}{2} \exp(-b|u|) \tag{44}$$

with Bernoulli parameter λ and scale parameter b, where δ is the Dirac distribution, admits the representation

$$p_U(u) = \int_{\mathbb{R}} \left(\sum_{v=0}^{1} p_{U \mid V=v, W=w}(u) p_V(v) \right) p_W(w)\, \mathrm{d}w, \tag{45}$$

where

$$p_V(v) = \lambda^{1-v} (1 - \lambda)^v \tag{46}$$

for $v \in \{0, 1\}$ is a Bernoulli distribution,

$$p_W(w) = \frac{b^2}{2} \exp\left(-\frac{b^2 w}{2} \right) \chi_{\mathbb{R}_{\geq 0}}(w) \tag{47}$$

[6] Sampling efficiency refers to effective samples per iteration; an efficiency of ρ means roughly $1/\rho$ iterations per "effective sample" (Gelman et al., 2013, Section 11.5).

is an exponential distribution, and

$$p_{U|V=v,W=w}(u) = \begin{cases} \delta(u) & \text{if } v = 0, \\ \text{Gauss}(0, w) & \text{if } v = 1. \end{cases} \tag{48}$$

The algorithm relies on the introduction of two latent vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$ that satisfy

$$p_{\mathbf{U}|\mathbf{V}=\mathbf{v},\mathbf{W}=\mathbf{w}}(\mathbf{u}) = \prod_{k=1}^{d} p_{U|V=\mathbf{v}_k,W=\mathbf{w}_k}(\mathbf{u}_k) \tag{49}$$

such that, as a result, the distribution conditioned on the measurements can be written as

$$\begin{aligned} p_{\mathbf{U},\mathbf{V},\mathbf{W}|\mathbf{Y}=\mathbf{y}}(\mathbf{u},\mathbf{v},\mathbf{w}) \propto \exp\!\big(-\tfrac{1}{2\sigma_{\text{n}}^2}\|\mathbf{H}\mathbf{u}-\mathbf{y}\|^2\big) \prod_{k=1}^{d} p_{U|V=\mathbf{v}_k,W=\mathbf{w}_k}(\mathbf{u}_k) \\ \times \prod_{k=1}^{d} \lambda^{1-\mathbf{v}_k}(1-\lambda)^{\mathbf{v}_k} \prod_{k=1}^{d} \frac{b^2}{2} \exp\!\left(-\frac{b^2\mathbf{w}_k}{2}\right), \end{aligned} \tag{50}$$

where $\mathbf{H} = \mathbf{A}\mathbf{D}^{-1}$. (48) and (50) imply that any sample from $p_{\mathbf{U}|\mathbf{V}=\mathbf{v},\mathbf{W}=\mathbf{w},\mathbf{Y}=\mathbf{y}}$ takes the value zero at those indices where $\mathbf{v}$ is zero, and values from a multivariate Gaussian distribution with covariance $\mathbf{C} = \big(\sigma_{\text{n}}^2\mathbf{H}\mathbf{H}^T + \text{diag}(\mathbf{w})\big)^{-1}$ and mean $\sigma_{\text{n}}^{-2}\mathbf{C}\mathbf{H}^T\mathbf{y}$ otherwise. Sampling $\mathbf{W} \mid \mathbf{U} = \mathbf{u}, \mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}$ amounts to the independent sampling of d one-dimensional distributions, which are $\text{Exp}(2/b^2)$ at those indices where $\mathbf{v}$ is zero and $\text{GIG}(b^2, \mathbf{u}_k^2, 0.5)$ those indices k where $\mathbf{v}$ is one. The conditional distribution of the binary support vector is

$$p_{\mathbf{V}|\mathbf{W}=\mathbf{w},\mathbf{Y}=\mathbf{y}}(\mathbf{v}) \propto |\mathbf{B}(\mathbf{v},\mathbf{w})|^{-\frac{1}{2}} \exp\!\big(-\tfrac{1}{2}\mathbf{y}^T\mathbf{B}(\mathbf{v},\mathbf{w})^{-1}\mathbf{y}\big) \prod_{k=1}^{d} \lambda^{1-\mathbf{v}_k}(1-\lambda)^{\mathbf{v}_k}, \tag{51}$$

where $\mathbf{B}(\mathbf{v},\mathbf{w}) = \sigma_{\text{n}}^2\mathbf{I} + \mathbf{H}\,\text{diag}(\mathbf{v} \odot \mathbf{w})\mathbf{H}^T$.[7] The standard way to sample from this distribution is to use a coordinate-wise Gibbs sampler that updates $\mathbf{v}_k \sim \text{Bernoulli}(p_k)$ with

$$p_k = (1 + \exp(-\Delta_k))^{-1} \tag{52}$$

where the log-odds increment

$$\begin{aligned} \Delta_k = \log\tfrac{1-\lambda}{\lambda} - \tfrac{1}{2}\big(\log|\mathbf{B}(\mathbf{v}_{k=1},\mathbf{w})| - \log|\mathbf{B}(\mathbf{v}_{k=0},\mathbf{w})|\big) \\ - \tfrac{1}{2}\big(\mathbf{y}^T\mathbf{B}(\mathbf{v}_{k=1},\mathbf{w})^{-1}\mathbf{y} - \mathbf{y}^T\mathbf{B}(\mathbf{v}_{k=0},\mathbf{w})^{-1})\mathbf{y}\big), \end{aligned} \tag{53}$$

where $\mathbf{v}_{k=\cdot} := (\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}, \cdot, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_d)$ is the difference between the log-posterior when the bit is on and when it is off. The resulting algorithm that is summarized in algorithm 4 can be interpreted as $(d+2)$-variable (*i.e.*, dimension-dependent) Gibbs method[8] and an efficient implementation is crucial.

D.2 Practical Gibbs Implementations

Sampling $\mathbf{X} \mid \mathbf{Z}$ in the GLM and $\mathbf{U} \mid \mathbf{V}, \mathbf{W}, \mathbf{Y}$ for the Bernoulli–Laplace case reduces to drawing from a high-dimensional Gaussian, which is a well-studied problem. For settings that necessitate a matrix-free implementation such as those that are commonly encountered in imaging applications, Kuric et al. (2025) advocate a Perturb-and-MAP sampler with preconditioned conjugate gradient solvers. For our moderate-dimensional problems with $d = 64$, a standard implementation based on the Cholesky factorization of the covariance matrix offered significantly faster (approximately one order of magnitude) sampling. The sampling of the different latent variables necessitates the sampling of the one-dimensional conditional latent distributions. All the conditional latent distributions that

[7] This is a different but equivalent formulation to what is presented by Bohra et al. (2023), who explicitly "slice" the matrices $\mathbf{H}$ and $\text{diag}(\mathbf{w})$ with the indices where $\mathbf{v}$ is one. We stick to this formulation since it requires less notation and emphasizes that implementations need not build variable-sized matrices, which is crucial for an efficient implementation on modern compute units that utilize highly parallelized computations.

[8] This is not strictly correct since the density violates the classical positivity conditions that are needed for Gibbs methods. It is a *partially collapsed* Gibbs method, see (Bohra et al., 2023; van Dyk & Park, 2008).

Algorithm 4 Bernoulli–Laplace Gibbs sampler.

Require: Initial increments $\mathbf{u}_0 \in \mathbb{R}^n$
for $s = 1, \dots, B + S$ **do**
  Draw $\mathbf{w}_i \sim p_{W|U=(\mathbf{u}_{s-1})_i, V=\mathbf{v}_i}$ ▷ *parallel over* i
  for $k = 1, \dots, d$ **do**
    Draw $\mathbf{v}_k \sim \text{Bernoulli}(p_k)$
  Draw $\mathbf{u}_s \sim p_{\mathbf{U}|\mathbf{V}=\mathbf{v},\mathbf{W}=\mathbf{w},\mathbf{Y}=\mathbf{y}}$
return $\{\mathbf{D}^{-1}\mathbf{u}_{B+s}\}_{s=1}^{S}$

are relevant in this paper admit efficient samplers that are readily available in standard scientific computing packages or can be implemented with little effort. We reuse the CUDA implementation of the generalized inverse Gaussian sampler from (Kuric et al., 2025) that implements the method proposed in (Devroye, 2012) and rely on `pytorch` (Paszke et al., 2017) for all others. Wherever possible, latent updates are parallelized.

In the Gibbs methods for the Bernoulli–Laplace increments, the sequential drawing of the binary support vector $\mathbf{V}$ is embedded in the outer Gibbs loop, which, in turn, may be embedded in the reverse-diffusion loop. This makes it crucial to minimize the use of heavy linear algebra operations to achieve acceptable runtimes. Writing $\mathbf{B}(\mathbf{v}, \mathbf{w}) = \sigma_{\text{n}}^2\mathbf{I} + \mathbf{H}\,\text{diag}(\mathbf{v} \odot \mathbf{w})\mathbf{H}^T$, we recognize that flipping the kth bit of $\mathbf{v}$ adds or removes a rank-one term $\mathbf{w}_k\mathbf{H}_k\mathbf{H}_k^T$, where $\mathbf{H}_k$ is the kth column of $\mathbf{H}$. Using the matrix determinant lemma and Woodbury–Sherman–Morrison, we update

$$\log|\mathbf{B}(\mathbf{v}_{k=1}, \mathbf{w})| = \log|\mathbf{B}(\mathbf{v}_{k=0}, \mathbf{w})| + \log(1 + \mathbf{w}_k\tau_k) \tag{54}$$

and

$$\mathbf{y}^T\mathbf{B}(\mathbf{v}_{k=1}, \mathbf{w})^{-1}\mathbf{y} = \mathbf{y}^T\mathbf{B}(\mathbf{v}_{k=0}, \mathbf{w})^{-1}\mathbf{y} - \frac{\mathbf{w}_k(\mathbf{H}_k^T\mathbf{B}(\mathbf{v}_{k=0}, \mathbf{w})^{-1}\mathbf{y})^2}{1 + \mathbf{w}_k\tau_k}, \tag{55}$$

where $\tau_k = \mathbf{H}_k^T\mathbf{B}(\mathbf{v}_{k=0}, \mathbf{w})^{-1}\mathbf{H}_k$. Thus, an efficient implementation factors $\mathbf{B}(\mathbf{v}, \mathbf{w})$ once per latent state, obtains the needed scalars via triangular solves, and performs rank-one updates as bits flip.

E DPS UPDATE STEPS

E.1 COVARIANCE IN C-DPS

C-DPS (Chung et al., 2023) uses the approximation of the likelihood

$$p_{\mathbf{Y}|\mathbf{X}_t=\mathbf{x}}(\mathbf{y}) \approx p_{\mathbf{Y}|\mathbf{X}_0=\mathbb{E}[\mathbf{X}_0|\mathbf{X}_t=\mathbf{x}]}(\mathbf{y}). \tag{56}$$

When the noise in the inverse problem is Gaussian, the likelihood score $\nabla\big(\mathbf{x} \mapsto \log p_{\mathbf{Y}|\mathbf{X}_0=\mathbb{E}[\mathbf{X}_0|\mathbf{X}_t=\mathbf{x}]}(\mathbf{y})\big)$ necessitates the computation of

$$\nabla\left(\mathbf{x} \mapsto \tfrac{1}{2}\|\mathbf{A}\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}] - \mathbf{y}\|^2\right), \tag{57}$$

which is

$$\mathbf{J}\left(\mathbf{x} \mapsto \mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]\right)(\,\cdot\,)\mathbf{A}^T(\mathbf{A}\mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \,\cdot\,] - \mathbf{y}) \tag{58}$$

after an application of the chain rule. The Jacobian $\mathbf{J}\left(\mathbf{x} \mapsto \mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]\right)$ is typically computed with automatic differentiation when $(\mathbf{x}, t) \mapsto \mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]$ is approximated with a neural network. In our framework, we use the connection with the covariance matrix $\text{Cov}[\mathbf{X}_0 \mid \mathbf{X}_t = \,\cdot\,]$. Indeed, as also shown in, *e.g.*, (Rissanen et al., 2025), if $\mathbf{X}_0$ and $\mathbf{X}_t$ verify (32), then

$$\tfrac{1}{1-\bar{\alpha}_t}\text{Cov}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}] = \tfrac{1}{\bar{\alpha}_t}\left(\mathbf{I} + (1-\bar{\alpha}_t)^2\nabla^2 \log p_{\mathbf{X}_t}(\mathbf{x})\right). \tag{59}$$

This identity combined with the derivative of (6) yields

$$\mathbf{J}\big(\mathbf{x} \mapsto \mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]\big)(\mathbf{x}_t) = \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\text{Cov}[\mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}_t]. \tag{60}$$

E.2 Explicit update steps

We define the instantiations of the update steps $\mathcal{S}(\mathbf{x}_t, \{\bar{\mathbf{x}}_s\}_{s=1}^S, \mathbf{y}, \mathbf{A}, \boldsymbol{\lambda}, t)$ with parameters $\boldsymbol{\lambda}$ for C-DPS, DiffPIR, and DPnP that are used in algorithm 2. Each $\mathbf{z}$ is a d-dimensional random vector with i.i.d. standard Gaussian entries.

C-DPS The input parameters $\boldsymbol{\lambda}$ are composed of the following. $\bar{\alpha}_t$ is the variance-preserving scaling weight as in (32), β_t is the variance of the diffusion transitions as in (35), and ζ parametrizes the likelihood-guidance strength. The diffusion noise level that corresponds to the denoising posterior is denoted $\sigma_t = (1-\bar{\alpha}_t)/\sqrt{\bar{\alpha}_t}$, which is used to compute the samples $\{\bar{\mathbf{x}}_s\}_{s=1}^S$.

$$
\begin{aligned}
\bar{\boldsymbol{\mu}} &= \frac{1}{S}\sum_{s=1}^{S} \bar{\mathbf{x}}_s \\
\mathbf{C} &= \frac{1}{S}\sum_{s=1}^{S} (\bar{\mathbf{x}}_s - \bar{\boldsymbol{\mu}})(\bar{\mathbf{x}}_s - \bar{\boldsymbol{\mu}})^T \\
\mathbf{x}'_{t-1} &= \tfrac{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \tfrac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\bar{\boldsymbol{\mu}} + \sigma_t \mathbf{z} \\
\tilde{\mathbf{x}}_{t-1} &= \mathbf{x}'_{t-1} - \tfrac{\zeta}{\|\mathbf{A}\bar{\boldsymbol{\mu}} - \mathbf{y}\|}\tfrac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathbf{C}^T\mathbf{A}^T(\mathbf{A}\bar{\boldsymbol{\mu}} - \mathbf{y}) \\
\mathbf{x}_{t-1} &= \tilde{\mathbf{x}}_{t-1}/\sqrt{\bar{\alpha}_{t-1}}
\end{aligned}
\tag{61}
$$

DiffPIR The input parameters are similar to the ones of C-DPS. $\bar{\alpha}_t$ and σ_t are defined in the same way as in C-DPS, and ζ parametrizes the likelihood-guidance strength. It also uses the noise level of the inverse problem σ_{n} and a balance hyperparamater γ.

$$
\begin{aligned}
\bar{\boldsymbol{\mu}} &= \frac{1}{S}\sum_{s=1}^{S} \bar{\mathbf{x}}_s \\
\rho_t &= \zeta\tfrac{\sigma_{\mathrm{n}}^2}{\sigma_t^2} \\
\bar{\mathbf{x}}_0 &= \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d}\big(\tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \tfrac{\rho_t}{2}\|\mathbf{x} - \bar{\boldsymbol{\mu}}\|^2\big) \\
\hat{\epsilon} &= \tfrac{1}{\sqrt{1-\bar{\alpha}_t}}\big(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\bar{\mathbf{x}}_0\big) \\
\tilde{\mathbf{x}}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\bar{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t-1}}(\sqrt{1-\gamma}\hat{\epsilon} + \sqrt{\gamma}\mathbf{z}) \\
\mathbf{x}_{t-1} &= \tilde{\mathbf{x}}_{t-1}/\sqrt{\bar{\alpha}_{t-1}}
\end{aligned}
\tag{62}
$$

DPnP The diffusion noise level that corresponds to the denoising posterior is denoted η_t, which is used to compute the sample $\bar{\mathbf{x}}_{s=1}$. This same η_t defines the likelihood-guidance strength.

$$
\begin{aligned}
\mathbf{x}_0 &= \bar{\mathbf{x}}_1 \\
\mathbf{x}_{t-1} &\sim \exp\Big(-\tfrac{1}{2}\|\mathbf{A}\cdot - \mathbf{y}\|^2 - \tfrac{1}{2\eta_t^2}\|\cdot - \mathbf{x}_0\|^2\Big)
\end{aligned}
\tag{63}
$$

The DPS template that is summarized in algorithm 2 is illustrated with a one-dimensional toy-example in fig. 4.

F Numerical Experiments

F.1 Forward Operators

We consider four forward operators in our experiments. First, the identity $\mathbf{A} = \mathbf{I} \in \mathbb{R}^{d\times d}$. This choice is motivated by the fundamental role that denoising algorithms currently play in many restoration algorithms and even labeling problems such as edge detection (Le et al., 2025). Second, a convolution operator $\mathbf{A} \in \mathbb{R}^{d\times d}$ that implements the convolution with a kernel that consists of the 13 central samples of a truncated Gaussian with variance 2 and is normalized to unit sum. We adopt circular boundary conditions to enable a fast computation of the proximal map that arises in the update step

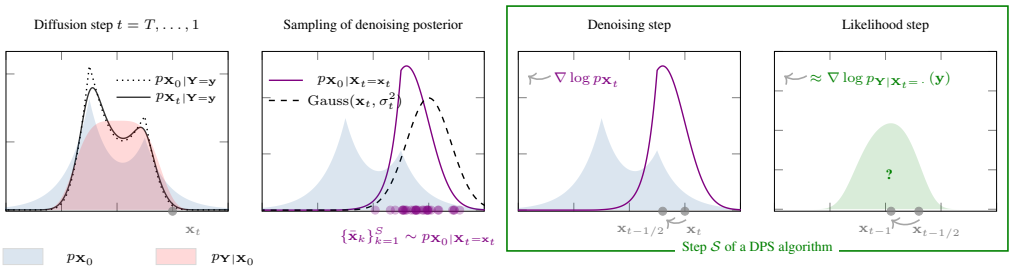

Figure 4: Illustration of the proposed template for DPS algorithms. The benchmarked posterior sampler targets $\mathbf{x}_0 \sim p_{\mathbf{X}_0|\mathbf{Y}=\mathbf{y}}$ via a diffusion process. At each diffusion time t, first the samples $\{\bar{\mathbf{x}}_k\}_{k=1}^S \sim p_{\mathbf{X}_0|\mathbf{X}_t=\mathbf{x}_t}$ are drawn from the denoising posterior. Then, the step $\mathcal{S}$ updates the iterate typically through a prior-guided update from the samples and a likelihood-guided update from the data. The likelihood guidance term is intractable and must be approximated, which constitutes the primary source of sampling error.

of DiffPIR (see appendix E.2) via the fast Fourier transform. Deconvolution is a relevant problem with applications in, *e.g.*, microscopy and astronomy. Third, a sampling operator $\mathbf{A} \in \mathbb{R}^{m \times d}$ that returns $m < d$ entries of its argument unchanged. This operator is also relevant in many fields such as image reconstruction and time-series forecasting. In particular, a forecasting or prediction problem can be modeled by returning the first m known entries recovering the remaining $(d - m)$ entries through the resolution of the inverse problem. In our experiments, each entry has an independent chance of $40\,\%$ being kept. Fourth, an operator $\mathbf{A} = \mathbf{MF} \in \mathbb{R}^{m \times d}$ where $\mathbf{F} \in \mathbb{R}^{2(\lfloor d/2 \rfloor + 1) \times d}$ is the matrix representation of the "real" one-dimensional discrete Fourier transform with separated real and imaginary components, and $\mathbf{M} \in \mathbb{R}^{m \times 2(\lfloor d/2 \rfloor + 1)}$ is a sampling operator. Such operators are relevant in, *e.g.*, medical imaging and astronomy. The sampling operator is constructed such that the 5 lowest frequencies (including the DC term) are acquired, and the remaining frequencies independently have a $40\,\%$ chance of being kept.

For all operators, the noise variance σ_{n}^2 is chosen such that the median measurement signal-to-noise ratio (SNR) is around $25\,\text{dB}$. We set $N_{\text{train}} = 1 \times 10^6$, $N_{\text{val}} = 1 \times 10^3$, and $N_{\text{test}} = 1 \times 10^3$.

F.2 Benchmark Implementation Details

The benchmarking pipeline starts with the generation of N_{test} test signals denoted $\{\mathbf{x}_k^{\text{test}}\}_{k=1}^{N_{\text{test}}}$ per increment distribution, each of which is independently synthesized by first drawing i.i.d. increments from the respective increment distribution and forming the signals via (42). It then proceeds to synthesize the N_{test} measurements (*i.e.* we use one noise instance per signal) denoted $\{\mathbf{y}_k^{\text{test}}\}_{k=1}^{N_{\text{test}}}$ according to (1) and, for each of the measurements, computes the gold-standard posterior samples of the various inverse problems via the Gibbs methods described in section 3. This stage is off-line (no reverse-diffusion loop) and trivially parallel across the measurements, which allows us to run long chains with burn-in periods of 1×10^5 iterations and obtain 2×10^5 draws from the posterior distribution. This far exceeds any values reported in (Kuric et al., 2025) or (Bohra et al., 2023) and results in precise MMSE estimates.

The dataset generation stage also involves the generation of N_{train} training signals $\{\mathbf{x}_k^{\text{train}}\}_{k=1}^{N_{\text{train}}}$ and N_{val} validation signals (mutually disjoint from the test signals) $\{\mathbf{x}_k^{\text{val}}\}_{k=1}^{N_{\text{val}}}$ and the corresponding validation measurements $\{\mathbf{y}_k^{\text{val}}\}_{k=1}^{N_{\text{val}}}$. The training signals are used for the learning of a neural score function like those that are used for the resolution of inverse problems when the prior is unknown or too expensive to evaluate. Training details are provided in appendix F.3 The validation signals are used to monitor the performance of the neural score function on unseen signals during the training stage and to tune the regularization parameters for the model-based approaches as well as the parameters of the DPS algorithms, see section 4.1 and section 4.1.

Unlike for the computation of the gold-standard MMSE estimate of the initial inverse problem, the denoising posteriors are sampled T times per trajectory (we use $T = 1000$). To ensure acceptable runtimes in this setting, we therefore pick the smallest burn-in period and sample count that still yield accurate estimates of the required statistics. We determine these settings with a rigorous protocol that

is detailed in appendix F.4. Ultimately, this protocol resulted in the choice of a burn-in period of 100 iterations and a sample count of 300.

F.3 Learning Details

For learned-based denoisers, a noise-conditional neural network with UNet architecture (305 761 learnable parameters) is trained in an off-line step on the N_{train} training signals in a standard setup (Adam optimizer with learning rate 1×10^{-4} with exponential decay with factor 0.9999, $100\,000$ parameter updates, batch size $10\,000$). The noise schedule in C-DPS and DiffPIR is defined by the two endpoints $\beta_0 = 1 \times 10^{-4}$ and $\beta_T = 2 \times 10^{-2}$ with linear equidistant samples in-between. The learned variant of DPnP is the "DDS-DDPM" variant (Xu & Chi, 2024, Algorithms 1 and 3) that contains an inner denoising-sampling loop. The oracle variant does not require an inner loop at all (except for the burn-in period), which makes the oracle variant the faster one for this case.

F.4 A Protocol to Determine the Burn-In Period and the Number of Samples

As discussed in appendix F.2, the burn-in period and the number of samples of the Gibbs samplers needs to be chosen appropriately to ensure an acceptable runtimes and a sufficiently small Monte Carlo error when they serve as the gold-standard samplers of the denoising posteriors that are encountered in the DPS algorithms. We determine the burn-in period and the number of samples through the following protocol that is run in an off-line stage prior to running the benchmark. We synthesize $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \mathbf{n}$ where σ_t is in the range defined by the noise schedule β, $\mathbf{x}_0$ is constructed via (42) for all four considered increment distrbutions, and $\mathbf{n}$ is some unknown but fixed vector of standard Gaussian noise. For each of the synthesized signals, we then launch $C = 1000$ parallel Gibbs chains on the corresponding denoising posterior and run those chains for $N_{\text{sufficient}}$ iterations, where $N_{\text{sufficient}}$ is a sufficiently large natural number that guarantees that the chains are stationary for at least N_{avg} (which is also relatively large) iterations and that, consequently, we can compute precise estimates of various statistics of the posterior distribution from the iterates from the last N_{avg} iterations across all C chains.

To determine the burn-in period, we then proceed to calculate a statistic that we can monitor throughout the iterations and that we can compare against the reference statistic. Specifically, denoting with $\mathbf{X}$ the random variable of the Gibbs sampler, we compute the empirical distribution of the increments at index 32, that is, $\mathbf{X}_{33} - \mathbf{X}_{32}$. The distribution of differences that is obtained by taking the last N_{avg} iterations across all C chains is considered the reference distribution. Then, we compute the Wasserstein-1 distance of that distribution to the one obtained by taking the average across N_{avg} iterations and all C in a sliding-window starting from the first Gibbs iterations. This allows us to gauge the burn-in period through a visual inspection of the Wasserstein-1 distance through the Gibbs iterations. In particular, we expect the Wasserstein-1 distance to be large for a number of initial samples where the Gibbs sampler is not stationary and then to oscillate around a small but nonzero value. The value will be nonzero due to the finite sample size. The Wasserstein-1 distance between the reference distribution and the one obtained through the Gibbs iterations is shown in fig. 5 (for the exemplary case of a $\mathrm{St}(1)$ distribution and a selection of noise variances). We observe that the empirical distribution of increments converges rapidly to the reference one. The Wasserstein-1 distance reaches the noise level after a single-digit number of iterations, which is in line with the analysis provided in (Kuric et al., 2025). Based on these findings, we chose the burn-in period as $B = 100$ iterations for all our experiments, which is more than sufficient to reach stationarity and has acceptable runtime.

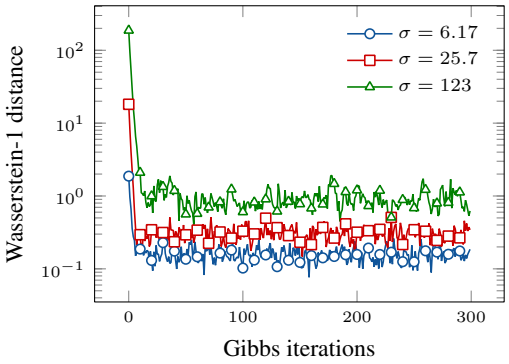

Figure 5: Wasserstein-1 distance of intermediate marginal distributions to that of the final sample.

To determine the number of samples that are needed for a sufficiently accurate computation of various statistics that any DPS algorithm may utilize in their update steps, we compute a precise estimation

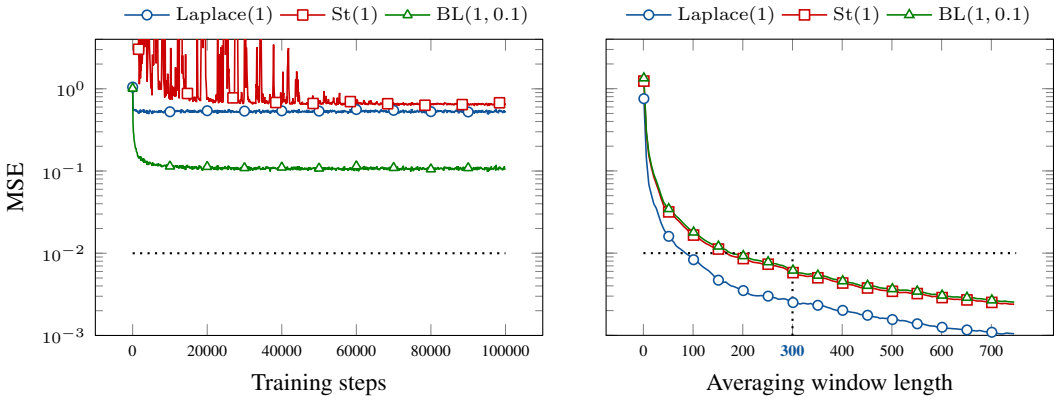

Figure 6: Left: Mean squared error normalized by the noise variance during the training of the learned denoiser. Right: Mean squared error between Monte Carlo estimates of the MMSE (with varying window lengths) and the long-run MMSE, normalized by the noise variance. In both plots, the dotted line indicates the agreed precision threshold that is reached at around 300 Gibbs iterations. To avoid clutter, we omit the curves for $\mathrm{Gauss}(0, 0.25)$, $\mathrm{St}(2)$, and $\mathrm{St}(3)$ that are qualitatively and quantitatively similar to the ones that are shown.

of the MMSE estimate of the denoising posterior by averaging the last N_{avg} iterations across all C chains. Then, we pick one arbitrary chain and grow a window from iteration $(N_{\text{avg}} - 1)$ to the left, average the samples in that window, and compute the MSE from the MMSE estimates obtained in the one-chain window to the precise estimate obtained by averaging the C chains and the last N_{avg} iterations. Motivated by the training loss of the neural denoisers, we pick a tolerance of 1×10^{-2} and monitor at which window length the MSE falls below that tolerance. The results in fig. 6 show that this tolerance is consistently reached when the averaging window is 300 samples long, which motivates our choice of using $S = 300$ samples for all our experiments.

F.5 Algorithm Parameter Identification

The adjustable regularization parameter for the method $\text{est} = \ell_2, \ell_1$ was found by

$$\lambda^{\text{est},\star} = \arg\min_{\lambda \in \Lambda} \frac{1}{N_{\text{val}}} \sum_{k=1}^{N_{\text{val}}} \tfrac{1}{d} \|\hat{\mathbf{x}}^{\text{est}}(\mathbf{y}_k^{\text{val}}, \lambda) - \mathbf{x}_k^{\text{val}}\|^2, \tag{64}$$

where Λ is the loglinear grid $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{N_{\text{mb}}}\}$ where

$$\lambda_i = 10^{a + (i-1)\frac{(b-a)}{N_{\text{mb}}-1}} \tag{65}$$

with $a = -5$ and $b = 5$. Since the model-based methods are very fast, we can use the relatively high $N_{\text{mb}} = 1000$.

The adjustable hyperparameters of the DPS methods were found by

$$\boldsymbol{\lambda}^{\text{alg},\star} = \arg\min_{\boldsymbol{\lambda} \in \boldsymbol{\Theta}^{\text{alg}}} \frac{1}{N_{\text{val}}} \sum_{k=1}^{N_{\text{val}}} \tfrac{1}{d} \|\hat{\mathbf{x}}^{\text{alg}}_{\text{MMSE}}(\mathbf{y}_k^{\text{val}}, \boldsymbol{\lambda}) - \mathbf{x}_k^{\text{val}}\|^2 \tag{66}$$

where the grid $\boldsymbol{\Theta}^{\text{alg}}$ is method-dependent. Note that this tuning is specifically tailored towards the evaluation with respect to the MMSE optimality gap. Due to resource constraints, the parameters are tuned for the learned denoiser. We use $N_{\text{samples}} = 10$ for the grid search on the validation set. We define a modest number of $N_{\text{dps}} = 40$ grid-points and found the extreme points of the grid (i.e. values of the parameters that clearly lead to worse results) by hand. For C-DPS and DiffPIR, we fix the diffusion schedule to standard choices ($\beta_0 = 1 \times 10^{-4}, \beta_T = 0.02$). In addition to the diffusion schedule, C-DPS has one tunable parameter γ that we tune on 40 loglinear grid points $(i = 1, \ldots, N_{\text{dps}})$

$$10^{a + (i-1)\frac{(b-a)}{N_{\text{dps}}-1}}, \tag{67}$$

where $a = -3$ and $b = 1$. DiffPIR has two tunable parameters γ and ζ, although γ is typically considered not so critical. Thus, we split the 40 grid points into a two-dimensional grid $\Theta^{\text{DiffPIR}} = \{0.3, 0.7\} \times \Theta^{\zeta}$, *i.e.*, 2 points for γ and 20 points for ζ given by $\Theta^{\zeta} = \{\Theta_1^{\zeta}, \dots, \Theta_{N_{\text{dps}}/2}^{\zeta}\}$ where

$$\Theta_1^{\zeta} = 10^{a+(i-1)\frac{(b-a)}{(N_{\text{dps}}/2)-1}} \tag{68}$$

with $a = -4$ and $b = 1$. The DPnP algorithm only has the schedule $\{\eta_t\}_{t=1}^{T}$ to tune. In this case, since DPnP is asymptotically correct, the schedule is a practical vehicle that enables to trade off between speed and accuracy. Therefore, we use a schedule that is similar to the one that was proposed in the original publication (Xu & Chi, 2024): We fix a small $\eta_{\text{final}} = 0.15$, and linearly decrease eta from some η_{initial} to η_{final} after $K/5$ initial iterations with η_{initial}:

$$\eta_i = \begin{cases} \eta_{\text{initial}} & \text{if} i = 1, \dots, K/5 \\ \frac{\eta_{\text{final}}}{\eta_{\text{initial}}}^{\frac{i-K/5}{K-K/5}} \eta_{\text{initial}} & \text{if} i = K/5+1, \dots, K \end{cases} \tag{69}$$

We treat η_{initial} as a tunable parameter and search over $\Theta^{\text{DPnP}} = \{\eta_1, \eta_2, \dots, \eta_{40}\}$ where for $i = 1, \dots, 40$,

$$\eta_i = 10^{a+(i-1)\frac{(b-a)}{40-1}} \tag{70}$$

with $a = -1$ and $b = 4$. Like in the original publication, we use the comparatively small $K = 40$.

The MSE over the validation data depending on the value of the adjustable regularization parameter of the ℓ_2 and ℓ_1 estimators and the adjustable hyperparameters of C-DPS, DiffPIR, and DPnP is shown in fig. 7. Since the γ parameter of DiffPIR is considered not so critical, we only show the values of the MSE for various choices of ζ where γ is set to the value of the optimal (γ, ζ) pair.

G Additional Results

An exhaustive quantitative evaluation of the change in the optimality gap when substituting the learned denoiser for the oracle denoiser is provided in table 4. The table also reports for which cases the oracle denoiser reports significantly better results than the learned denoiser according to a Wilcoxon signed-rank test ($p = 0.05$, N_{test} pairs, two-sided test with the winner determined by the median of differences). We attribute a better performance of the learned denoiser to the fact that the algorithms are fine-tuned using the learned component or to the cases where the likelihood score approximation is compensated by the one of the learned component.

We show uncurated qualitative results of the MMSE estimate obtained by the DPS algorithms and the gold-standard Gibbs methods in figs. 8 to 15. The figures alternate between the oracle denoiser and the learned denoiser and show the results for deconvolution, denoising, imputation, and reconstruction from partial Fourier samples in that order. Each figure contains results for $\text{BL}(0.1, 1)$, $\text{St}(1)$, $\text{St}(2)$, and $\text{Laplace}(1)$.

The coverage results for $\alpha = 0.9$ are presented in table 5. The Gibbs row again validates the implementation; for all forward operators, they achieve coverages that are very close to 0.9. In contrast, the coverage values obtained by the DPS algorithms are generally much smaller than 0.9. For C-DPS and DiffPIR, the reported coverage values are almost always 0 except for $\text{BL}(0.1, 1)$ and $\text{St}(1)$ increments, where the coverages are usually (close to) 1 for C-DPS and inconsistent for DiffPIR. For almost all increment distributions and forward operators, DPnP reports coverage values that are closest to but typically smaller than 0.9. Note that a coverage of 1 can be considered the worst case even at a target of 0.9. For instance, it is achieved by setting all samples to a constant vector with extremely large (i.e. "unlikely") entries.

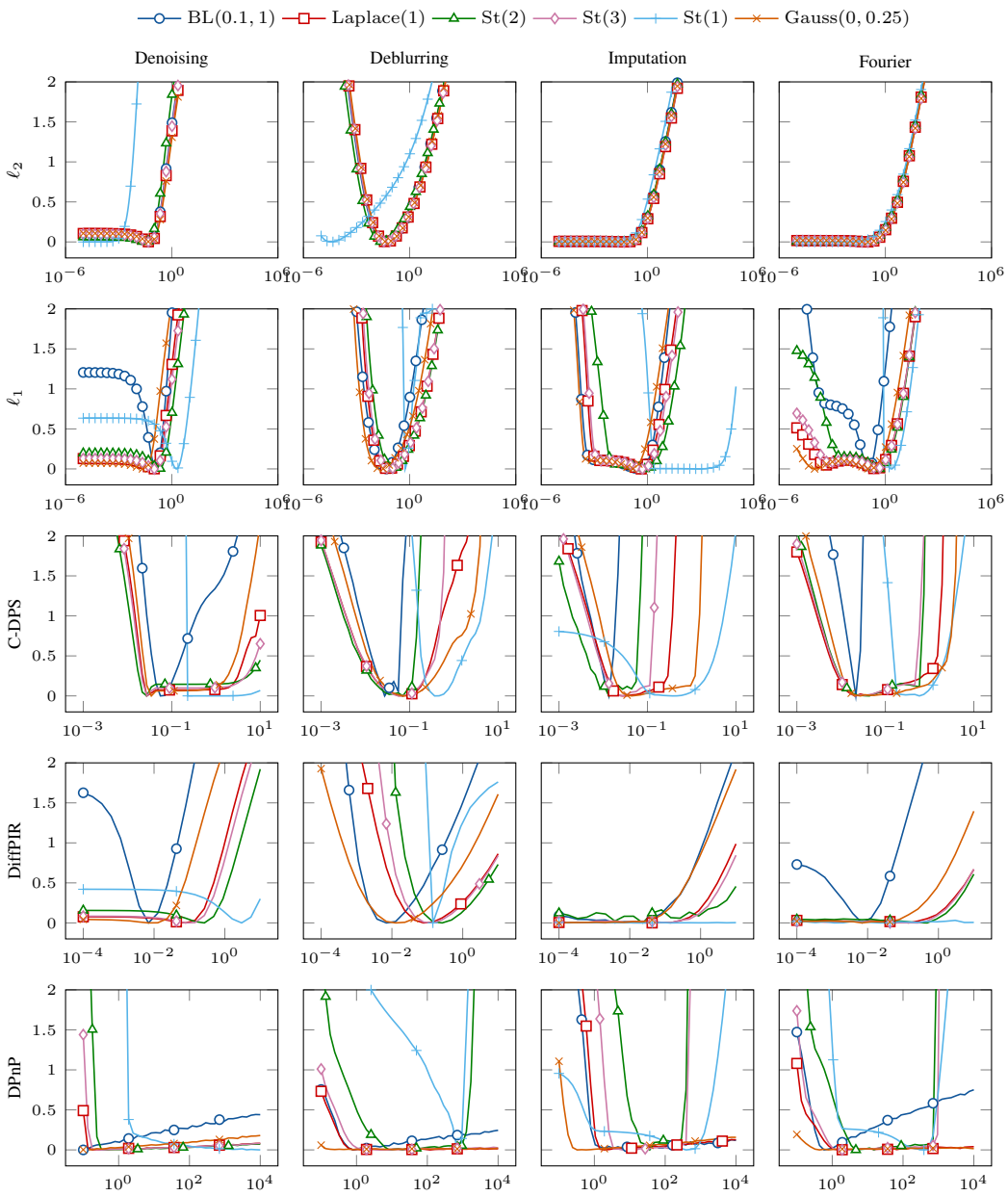

Figure 7: Grid search diagnostics (logarithm of the MSE over the validation data set) for the model-based methods and the DPS algorithms. Rows: ℓ_2; ℓ_1; C-DPS; DiffPIR; DPnP. Columns: Denoising; deconvolution; imputation; reconstruction from partial Fourier measurements. For better visualization, each curve has had its minimum subtracted. To avoid clutter, marks are placed only at every 10th grid point.

Table 4: Change in MMSE optimality gap (mean $\pm$ standard deviation) after substituting the learned denoiser with the oracle denoiser. An asterisk indicates a significant changes according to a Wilcoxon signed-rank test ($p = 0.05$). Negative number with asterisk: MMSE estimates obtained with the oracle denoiser are significantly better. Positive number with asterisk: MMSE estimates obtained with the learned denoiser are significantly better.

		Gauss(0.25)	Laplace(1)	BL(0.1, 1)	St(1)	St(2)	St(3)
Denoising	C-DPS	0.00 ± 0.11	0.00 ± 0.16	$-0.46 \pm 1.16^*$	0.00 ± 0.01	$0.02 \pm 0.79^*$	-0.01 ± 0.14
	DiffPIR	0.00 ± 0.13	0.00 ± 0.17	$-0.05 \pm 0.78^*$	$-0.41 \pm 0.80^*$	0.00 ± 0.20	0.00 ± 0.15
	DPnP	$0.04 \pm 0.27^*$	-0.01 ± 0.22	$-0.55 \pm 1.31^*$	$-0.77 \pm 1.31^*$	0.00 ± 0.24	0.00 ± 0.23
Deconvolution	C-DPS	-0.01 ± 0.24	0.00 ± 0.26	$0.09 \pm 0.97^*$	$6.64 \pm 3.21^*$	$-0.12 \pm 1.11^*$	-0.03 ± 0.43
	DiffPIR	-0.01 ± 0.23	0.00 ± 0.23	0.04 ± 1.12	$13.56 \pm 9.90^*$	-0.01 ± 0.47	0.00 ± 0.31
	DPnP	0.00 ± 0.25	$-0.01 \pm 0.27^*$	-0.02 ± 1.20	$-4.98 \pm 3.86^*$	0.06 ± 0.77	-0.02 ± 0.34
Imputation	C-DPS	0.00 ± 0.30	0.01 ± 0.35	$0.41 \pm 1.51^*$	$3.41 \pm 4.99^*$	$-0.12 \pm 1.01^*$	-0.01 ± 0.57
	DiffPIR	0.00 ± 0.29	0.00 ± 0.33	0.03 ± 1.05	$-0.20 \pm 3.05^*$	0.03 ± 0.71	0.00 ± 0.47
	DPnP	0.00 ± 0.35	-0.02 ± 0.38	-0.02 ± 1.02	$-10.46 \pm 5.70^*$	0.02 ± 0.67	-0.01 ± 0.48
Fourier	C-DPS	-0.02 ± 0.43	-0.01 ± 0.49	$0.80 \pm 1.43^*$	$0.09 \pm 5.63^*$	$-0.03 \pm 0.79^*$	0.01 ± 0.49
	DiffPIR	-0.01 ± 0.39	0.00 ± 0.40	$0.12 \pm 0.83^*$	$-0.64 \pm 1.70^*$	$-0.03 \pm 0.42^*$	-0.02 ± 0.38
	DPnP	-0.01 ± 0.43	0.00 ± 0.45	$-0.33 \pm 1.13^*$	$-1.32 \pm 3.18^*$	0.00 ± 0.54	0.01 ± 0.46

Table 5: Posterior coverage of various estimation methods at $\alpha = 0.9$.

		Gauss$(0, 0.25)$		Laplace(1)		BL(0.1, 1)		St(1)		St(2)		St(3)	
		Learned	Oracle	Learned	Oracle	Learned	Oracle	Learned	Oracle	Learned	Oracle	Learned	Oracle
Denoising	Gibbs	—	0.90	—	0.91	—	0.91	—	0.89	—	0.91	—	0.89
	C-DPS	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
	DiffPIR	0.00	0.00	0.00	0.00	1.00	1.00	0.28	0.02	0.00	0.00	0.00	0.00
	DPnP	0.58	0.67	0.11	0.11	1.00	0.41	0.53	0.08	0.09	0.09	0.09	0.10
Deconvolution	Gibbs	—	0.89	—	0.90	—	0.90	—	0.91	—	0.91	—	0.91
	C-DPS	0.00	0.00	0.01	0.00	1.00	1.00	1.00	0.83	0.01	0.00	0.00	0.00
	DiffPIR	0.00	0.00	0.00	0.00	1.00	1.00	0.97	0.92	0.00	0.00	0.00	0.00
	DPnP	0.12	0.12	0.06	0.07	1.00	0.31	0.50	0.06	0.06	0.06	0.07	0.06
Imputation	Gibbs	—	0.89	—	0.90	—	0.86	—	0.91	—	0.91	—	0.91
	C-DPS	0.00	0.00	0.00	0.00	1.00	1.00	0.94	0.78	0.15	0.15	0.00	0.00
	DiffPIR	0.00	0.00	0.00	0.00	1.00	1.00	0.72	0.32	0.00	0.00	0.00	0.00
	DPnP	0.28	0.31	0.09	0.08	1.00	0.41	0.56	0.07	0.14	0.13	0.12	0.13
Fourier	Gibbs	—	0.91	—	0.90	—	0.90	—	0.91	—	0.92	—	0.91
	C-DPS	0.00	0.00	0.00	0.00	1.00	1.00	0.96	0.74	0.01	0.01	0.00	0.00
	DiffPIR	0.00	0.00	0.00	0.00	1.00	1.00	0.92	0.65	0.00	0.01	0.00	0.00
	DPnP	0.19	0.19	0.08	0.06	1.00	0.32	0.50	0.06	0.07	0.07	0.07	0.06

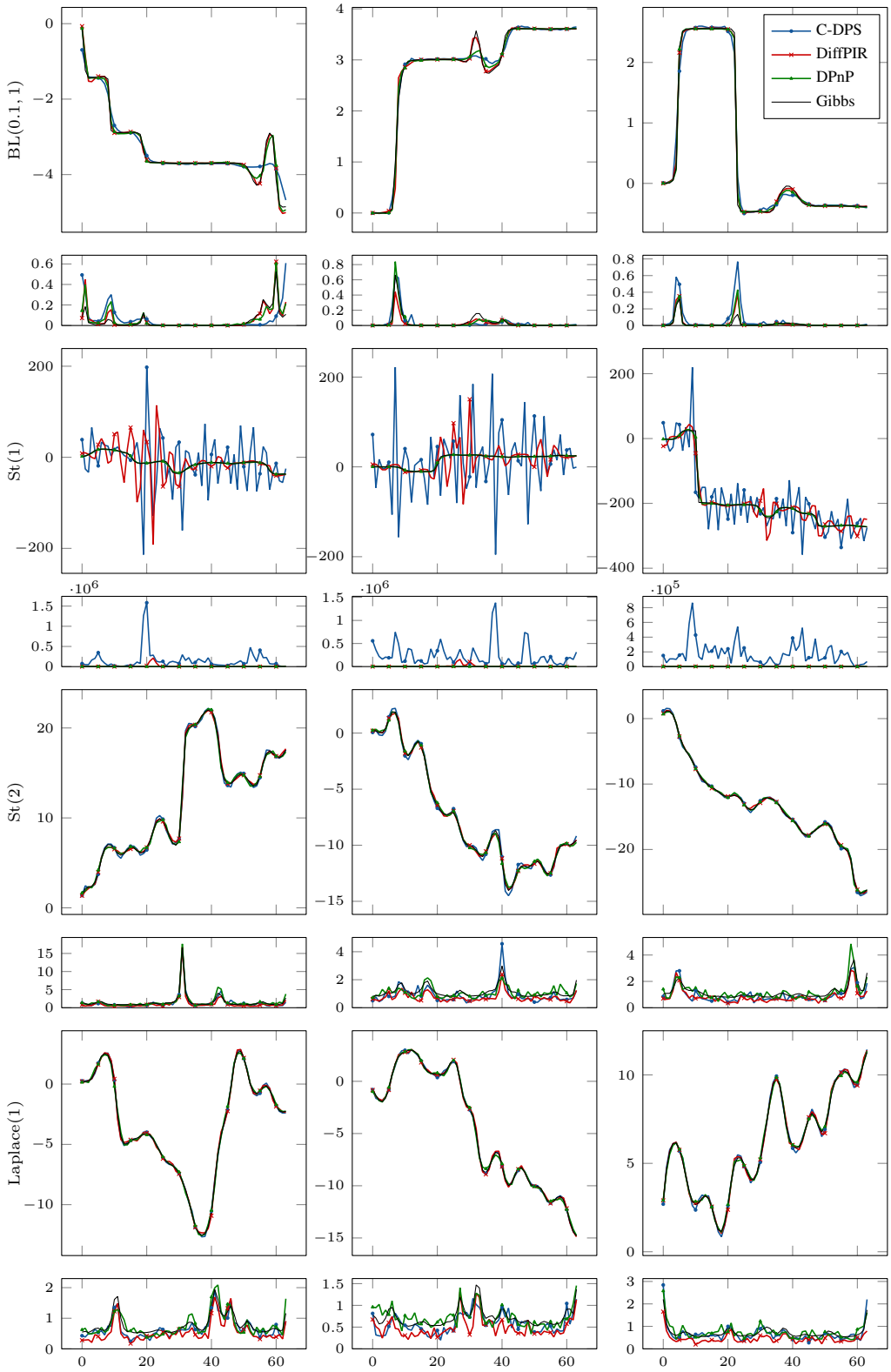

Figure 8: Qualitative results for deconvolution using the oracle denoising sampler. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

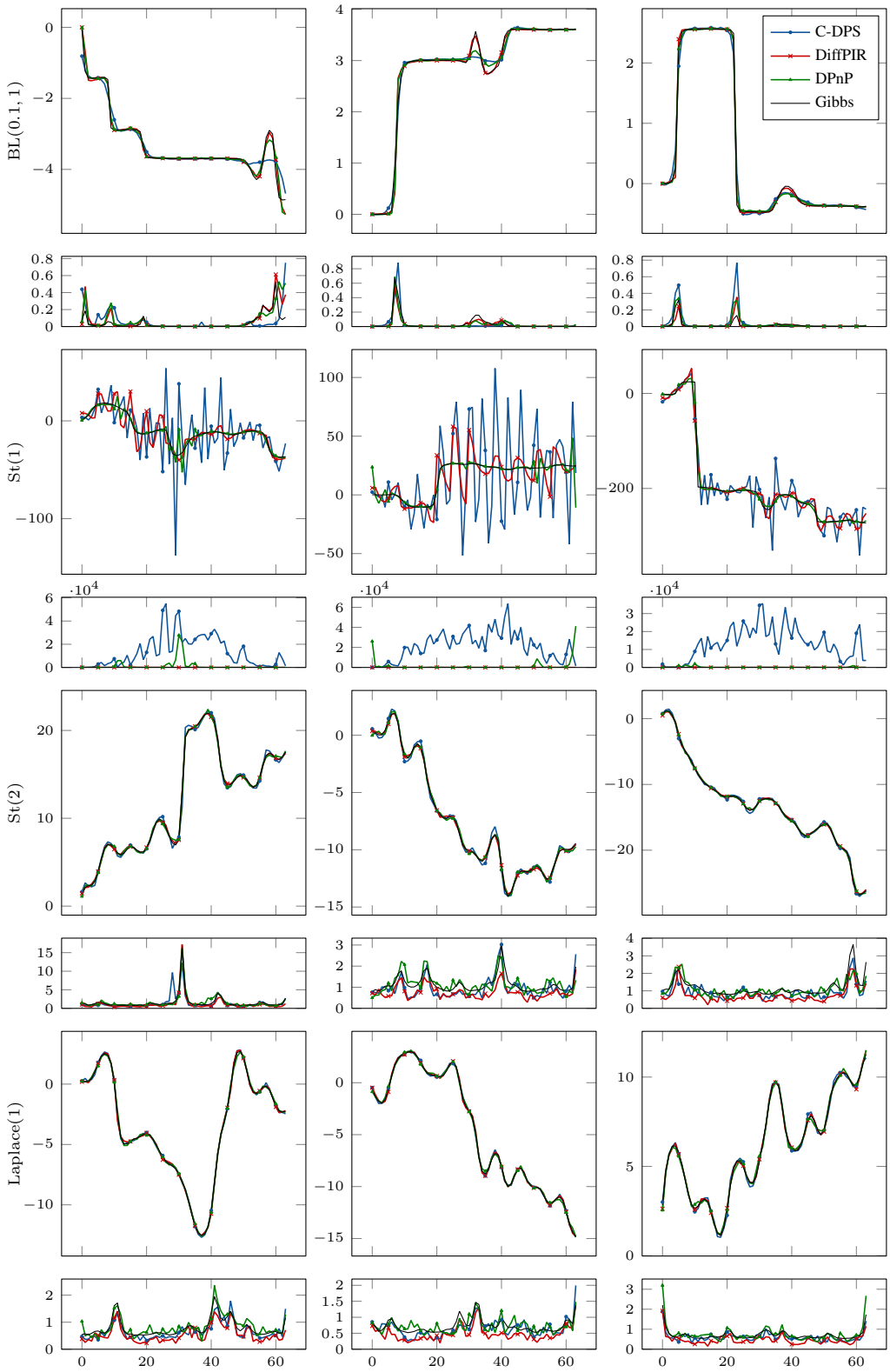

Figure 9: Qualitative results for deconvolution using the learned denoiser. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

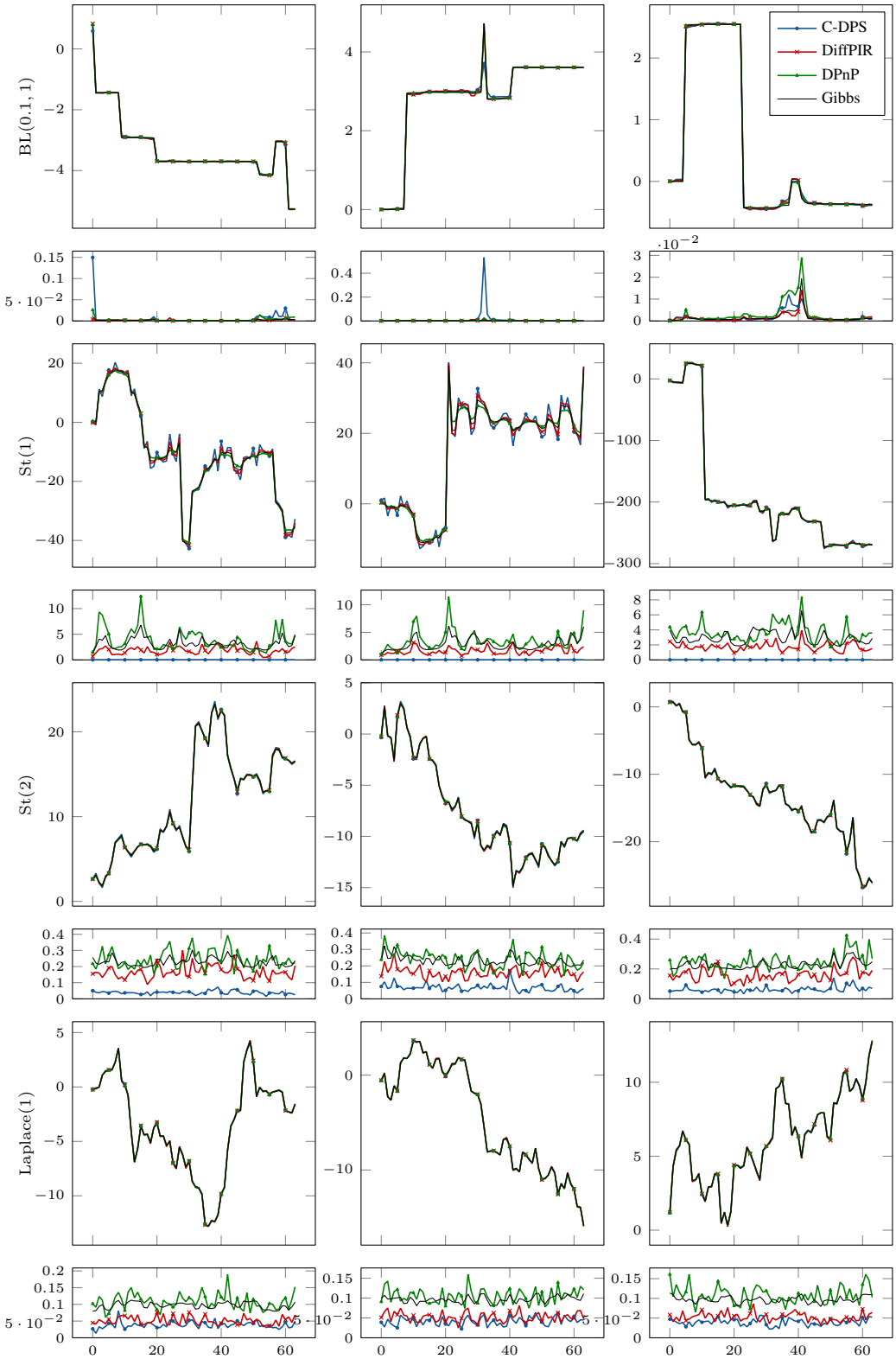

Figure 10: Qualitative results for denoising using the oracle MMSE denoiser. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

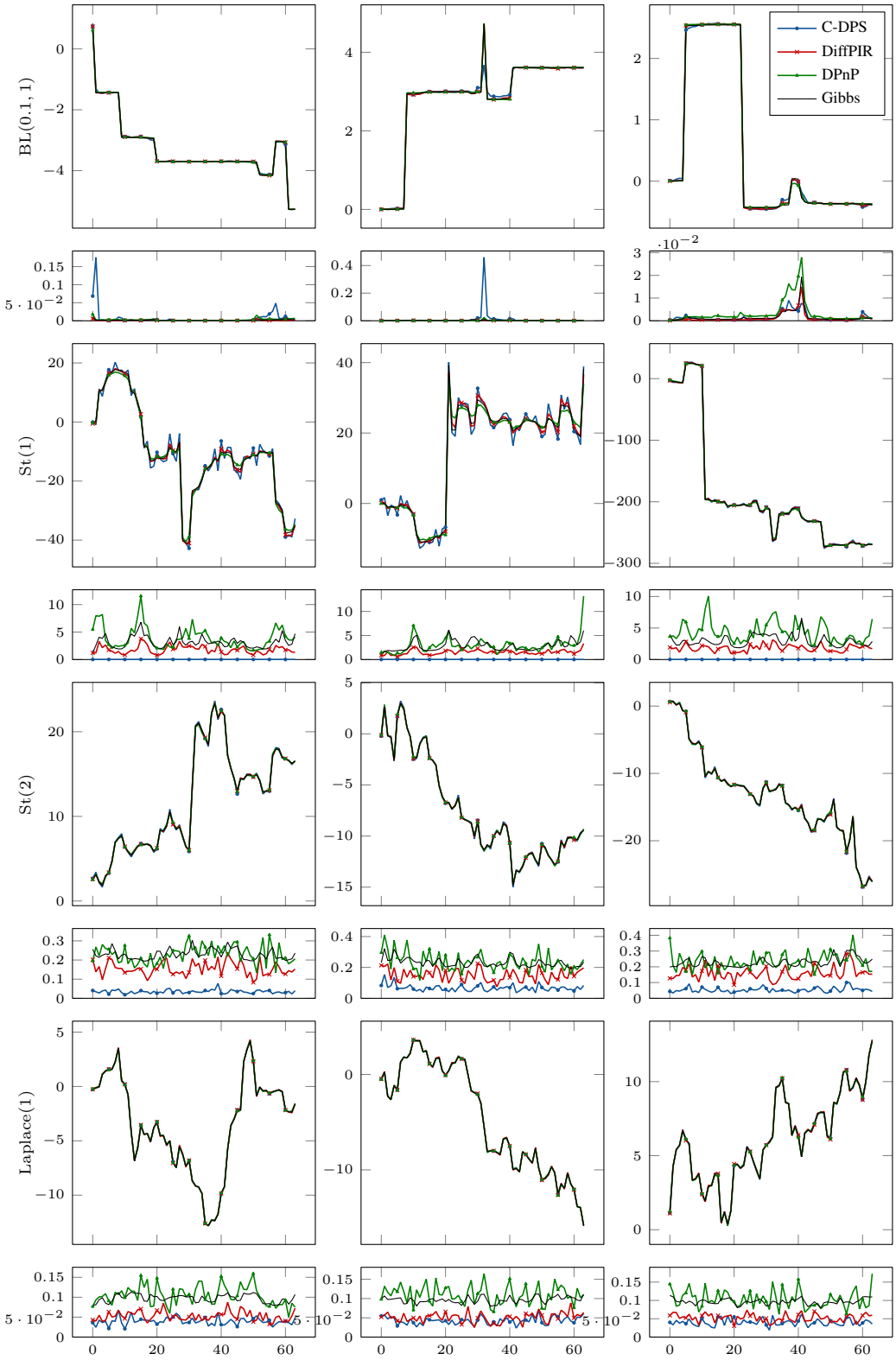

Figure 11: Qualitative results for denoising using the learned denoiser. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

Figure 12: Qualitative results for imputation using the oracle sampler. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

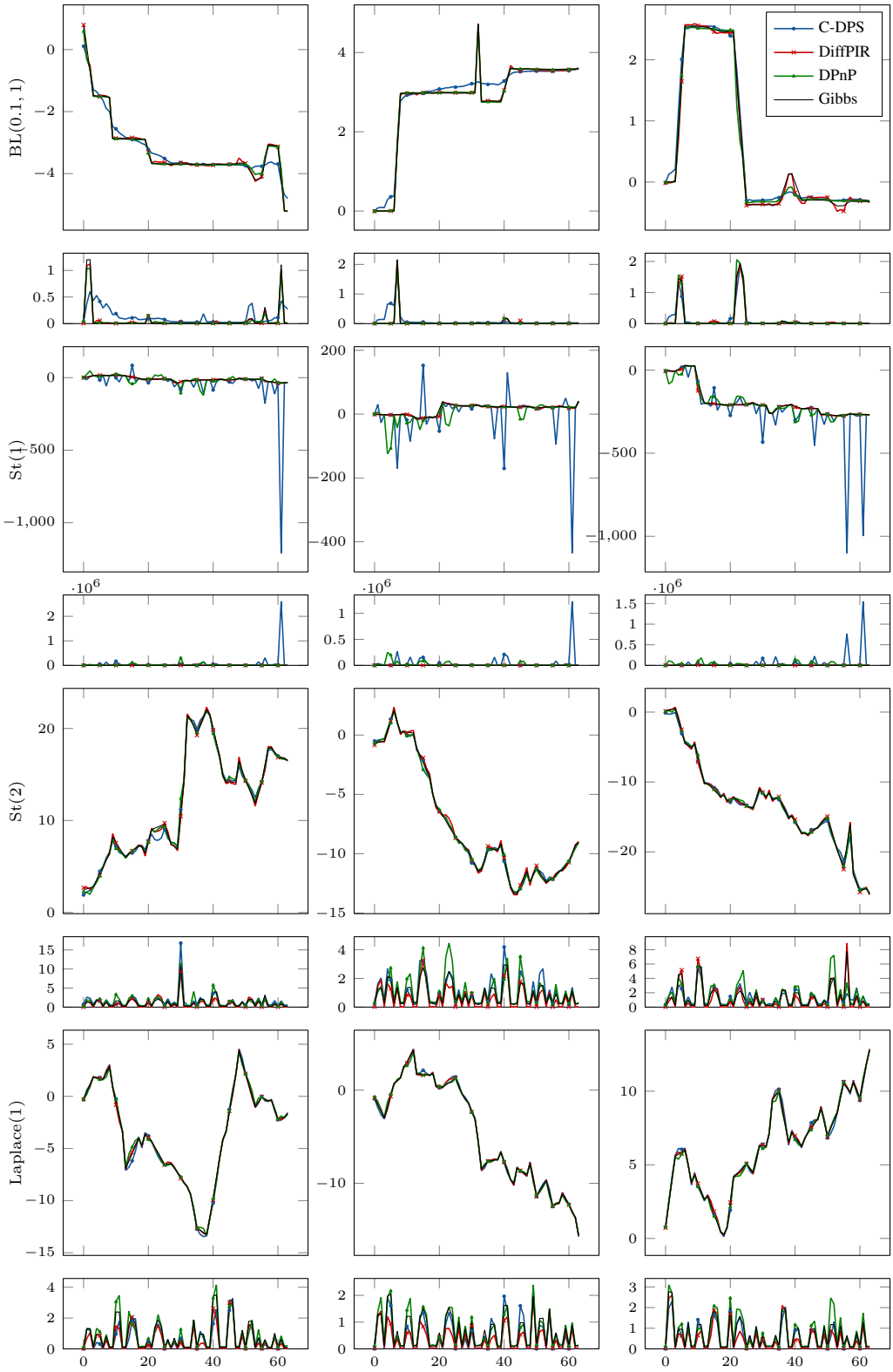

Figure 13: Qualitative results for imputation using the learned denoiser. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

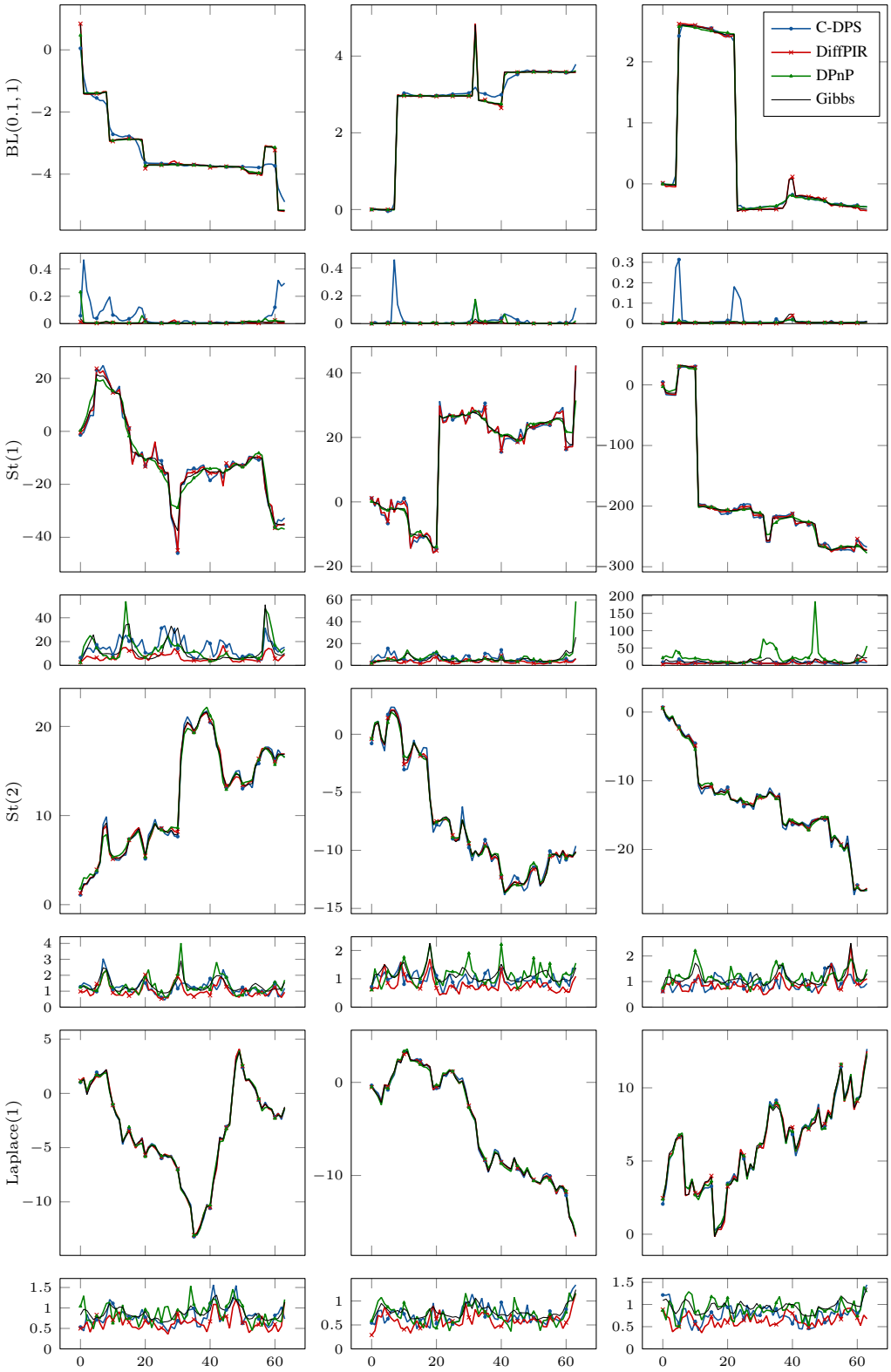

Figure 14: Qualitative results for reconstruction from partial Fourier measurements using the oracle denoiser. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

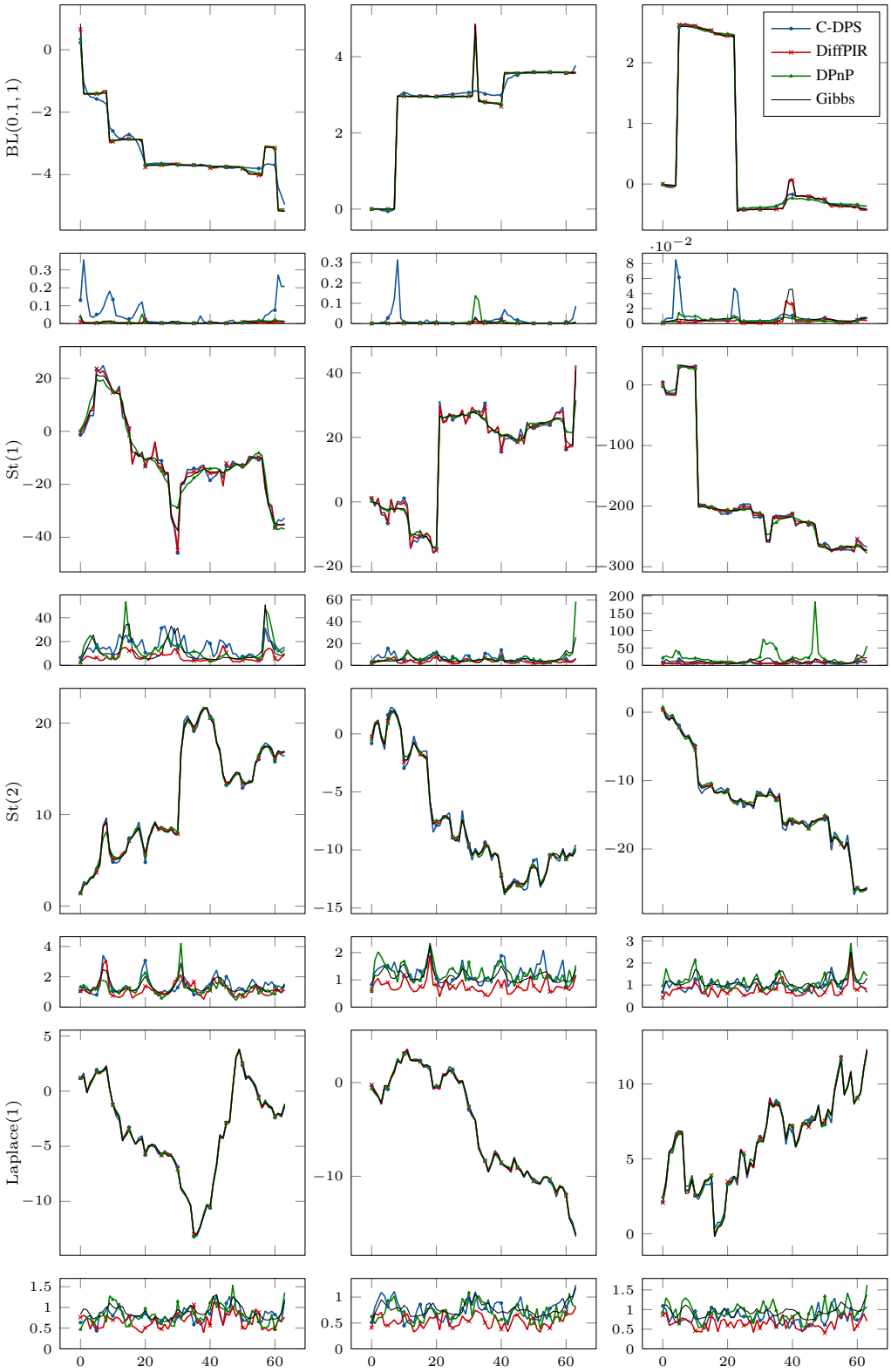

Figure 15: Qualitative results for reconstruction from partial Fourier measurements using the learned denoiser. Rows: increment distributions. For each increment distribution, the MMSE estimates obtained by the different DPS algorithms and the gold-standard Gibbs methods are shown on top of the corresponding index-wise marginal variances. Columns: Different measurements.

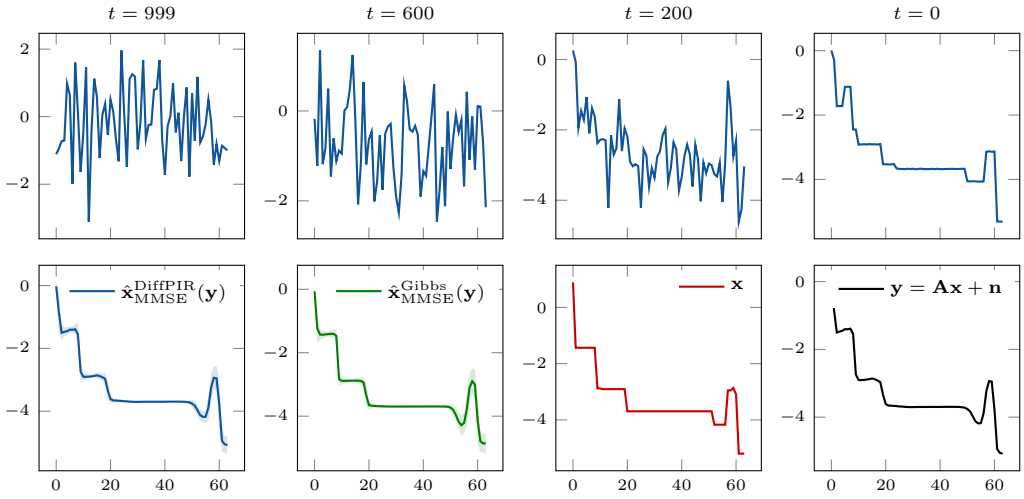

Figure 16: Conditional generation for deconvolution of a signal with $\mathrm{BL}(0.1, 1)$ increments with DiffPIR. Top: Prototypical sampling trajectory at times $t = 999, 600, 200, 0$. Bottom: From left to right: MMSE estimate obtained by averaging all DiffPIR samples; gold-standard MMSE estimate obtained by the Gibbs method; the data-generating signal; the data.