KALMAN POSTERIOR SAMPLING WITH DIFFUSION PRIORS AND SIMULATION-BASED LIKELIHOODS

Anonymous authors
Paper under double-blind review

ABSTRACT

In science, measurements are often modeled by stochastic numerical simulators whose randomness is non-Gaussian and whose likelihood and gradients are rarely available. Diffusion posterior sampling usually assumes all three, which rules these simulators out. We introduce Kalman Posterior Sampling (KPS), a sampler in which the diffusion prior and a statistical linearization of the likelihood defined by the simulator enter a common Kalman update through their moments, using simulations alone or gradients when the simulator is differentiable. An inner Gibbs loop keeps the iterates consistent with the diffusion process, preventing errors from accumulating across steps. On latent fluid dynamics and scientific inverse problem benchmarks, KPS outperforms state-of-the-art samplers in accuracy with as few as 16 particles, delivers close-to-calibrated uncertainty, and captures multimodal posteriors even from black-box simulators.

1 INTRODUCTION

In many scientific problems, quantities of interest, such as the state of a physical system or model parameters, are only accessible indirectly through observations. This is, for example, the case in meteorology, where temperature profiles are indirectly observed from satellite radiance measurements (Bouillon et al., 2022; Pinnington et al., 2026). The problem to solve, commonly referred to as *inverse problem*, therefore consists of recovering the quantity of interest from the observation.

However, observations often provide only partial information about the quantity of interest, leaving multiple solutions consistent with the same observation (Baumeister, 1987). This naturally leads to a Bayesian formulation of inverse problems: given a prior distribution $p(x)$ over the quantity of interest and an observation y of the latter, we seek to sample from the posterior distribution $p(x \mid y)$, that is, the distribution of plausible solutions given the observation (Dashti & Stuart, 2017). This task, known as *posterior sampling*, provides a probabilistic formulation of inverse problems.

Traditionally, the prior distribution $p(x)$ is modeled using simple parametric distributions (Marnissi et al., 2017), but recent work has increasingly turned to diffusion-based generative models (Song et al., 2021; Ho et al., 2020). Indeed, these models have proven to be powerful tools for learning high-dimensional distributions from data (Podell et al., 2024), and various algorithms have consequently been proposed to adapt them for inverse problems without further training (Chung et al., 2023; Wu et al., 2023). Motivated by these advances, we propose a method for adapting such models to scientific inverse problems, where a flexible formulation (Section 1.1) is desirable to cover a broad range of problems across scientific domains (Zheng et al., 2025b; Deistler et al., 2025).

1.1 PROBLEM STATEMENT

Given an observation $y \in \mathbb{R}^d$ and a prior distribution $p(x)$ over the variable of interest $x \in \mathbb{R}^n$, the goal is to sample from the posterior distribution $p(x \mid y)$. We assume access to a trained diffusion model (Section 2) that samples from $p(x)$, as well as a forward operator $\mathcal{A}$ modeling the relationship between the observation and the quantity of interest

$$y = \mathcal{A}(x, \eta), \qquad \eta \sim p(\eta), \tag{1}$$

where $\eta \in \mathbb{R}^m$ is a stochastic term capturing sources of uncertainty, such as measurement errors. This general relationship encompasses simpler formulations commonly found in the posterior sam-

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

pling literature, such as the additive Gaussian model (Chung et al., 2023; Zheng et al., 2025a)

$$y = \mathcal{A}(x) + \eta, \qquad \eta \sim \mathcal{N}(0, \Sigma_y), \tag{2}$$

but also covers more complex scientific problems, such as black-hole imaging (Zheng et al., 2025b) and simulation-based inference applications (Cranmer et al., 2020). In particular, since the likelihood induced by the general formulation of Equation 1 and defined as

$$p(y \mid x) = \int_{\mathbb{R}^m} \delta\big(y - \mathcal{A}(x, \eta)\big) p(\eta) \, \mathrm{d}\eta \tag{3}$$

is intractable without simplifying assumptions on $\mathcal{A}$, it covers likelihood-free scientific problems.

1.2 Contributions

We propose Kalman Posterior Sampling (KPS), a posterior sampling algorithm for trained diffusion priors (Section 2). Unlike existing methods (Section 3), KPS does not require access to the likelihood and, in its hybrid variant, also requires no gradients (Section 4). Using posterior-level metrics (Sharief et al., 2026; Lemos et al., 2023), we show that KPS matches or improves upon existing samplers on a latent diffusion problem. On InverseBench, KPS-Gradient is close to calibrated, while KPS-Hybrid is under-dispersed but improves as the ensemble size increases (Section 5).

2 Diffusion priors

The primary purpose of diffusion models (Ho et al., 2020), also known as score-based generative models (Song et al., 2021), is to generate plausible data from a distribution $p(x)$ of interest. Formally, adapting the formulation of Song et al. (2021) and the notation of Rozet et al. (2024), samples $x \in \mathbb{R}^n$ from $p(x)$ are progressively perturbed through a diffusion process expressed as a stochastic differential equation (SDE)

$$\mathrm{d}x_t = f_t x_t \mathrm{d}t + g_t \mathrm{d}w_t, \tag{4}$$

where $f_t \in \mathbb{R}$ is the drift coefficient, $g_t \in \mathbb{R}_+$ the diffusion coefficient, $w_t \in \mathbb{R}^n$ a standard Wiener process, and $x_t \in \mathbb{R}^n$ the perturbed sample at time $t \in [0, 1]$. Because the SDE is linear, the perturbation kernel from x to x_t is Gaussian and takes the form

$$p(x_t \mid x) = \mathcal{N}(x_t \mid \alpha_t x, \Sigma_t), \tag{5}$$

where α_t and $\Sigma_t = \sigma_t^2 I$ are derived from f_t and g_t (Lai et al., 2025). Crucially, the forward SDE of Equation 4 has an associated family of reverse SDEs (Anderson, 1982)

$$\mathrm{d}x_t = \left[f_t x_t - \frac{1 + \lambda^2}{2} \, g_t^2 \nabla_{x_t} \log p_t(x_t) \right] \mathrm{d}t + \lambda \, g_t \, \mathrm{d}w_t, \tag{6}$$

where λ is a parameter controlling stochasticity. In simpler words, we can draw noise samples from $p(x_1) \approx \mathcal{N}(0, \Sigma_1)$ and gradually remove the noise to obtain samples from $p(x)$ by simulating Equation 6 from $t = 1$ to $t = 0$ using an appropriate discretization scheme Zhang & Chen (2023).

In practice, the score $\nabla_{x_t} \log p(x_t)$ in Equation 6 is unknown, but can be approximated by a neural network d_θ called denoiser and trained by minimizing $\mathbb{E}_{x,t,x_t}\left[w_t \| x - d_\theta(t, x_t) \|_2^2\right]$, whose optimum $\mathbb{E}[x \mid x_t]$ is linked to the score through the first order Tweedie's formula (Tweedie, 1947) as

$$\nabla_{x_t} \log p(x_t) = \Sigma_t^{-1} (\alpha_t \mathbb{E}[x \mid x_t] - x_t). \tag{7}$$

Similar formulations have also been proposed, including flow matching (Lipman et al., 2023) and stochastic interpolants (Albergo et al., 2025). However, when the reference distribution $p(x_1)$ is Gaussian, these formulations are theoretically equivalent, so what follows applies to them as well.

3 Related Work

As discussed in Section 1, several algorithms have been proposed to solve inverse problems using trained diffusion priors. However, most of them were developed for the additive Gaussian model of Equation 2, and do not directly extend to complex scientific problems with non-differentiable forward operator or intractable likelihoods. These methods, which serve as baselines in our experiments (Section 5), are summarized in Table 1 along with their degree of flexibility. The main ideas behind these algorithms are also discussed below, with a more complete description in Appendix E.

Guidance-based methods Replacing the score of the prior in Equation 6 with the score of the posterior yields samples from the target posterior distribution $p(x \mid y)$ (Song et al., 2021). This score is unknown a priori, but can be decomposed using Bayes' rule as

$$\nabla_{x_t} \log p_t(x_t \mid y) = \underbrace{\nabla_{x_t} \log p_t(x_t)}_{\text{prior score}} + \underbrace{\nabla_{x_t} \log p(y \mid x_t)}_{\text{likelihood score}}. \tag{8}$$

The prior score is available from the trained model, leaving the likelihood score to be estimated. However, this score is intractable in practice because the likelihood $p(y \mid x_t)$ involves integrating over the denoising posterior $p(x \mid x_t)$:

$$p(y \mid x_t) = \int p(y \mid x) p(x \mid x_t) \, \mathrm{d}x = \mathbb{E}_{x \sim p(x \mid x_t)} \left[p(y \mid x) \right]. \tag{9}$$

Existing approaches address this issue by introducing an approximation of $p(x \mid x_t)$ (Chung et al., 2023; Rozet et al., 2024; Zheng et al., 2025a), which provides a tractable estimate of $p(y \mid x_t)$ and allows the likelihood score to be computed by automatic differentiation. More recent methods based on stochastic flow maps (Holderrieth et al., 2026; Potaptchik et al., 2026; McCallum et al., 2026) instead directly approximate the expectation in Equation 9 by Monte Carlo, using a map $\Phi_\theta(\varepsilon; t, x_t)$ that generates samples from $p(x \mid x_t)$ in a single function evaluation.

Sequential Monte-Carlo Rather than approximating the likelihood score, Sequential Monte Carlo (SMC) methods (Wu et al., 2023; Dou & Song, 2024) propagate weighted particles using an approximate posterior score, followed by reweighting and resampling at each step. While asymptotically exact, these methods scale poorly with the dimensionality of the problem.

Variable splitting Methods such as DAPS (Zhang et al., 2025) and PnP-DM (Wu et al., 2024) split the posterior inference into two separate steps. One step samples the prior distribution by solving Equation 6, while the other step enforces consistency with the observation through Langevin dynamics. The two steps are repeated successively, progressively moving toward the target posterior. PPR (Rochman-Sharabi & Louppe, 2026) follows the same alternation to enforce hard constraints, projecting through the denoiser and re-noising through the forward kernel.

Ensemble Kalman EnKG (Zheng et al., 2025a) estimates the likelihood score from denoised particles pushed through the forward model, in the spirit of ensemble Kalman inversion (Iglesias et al., 2013), which removes the need for gradients but targets a MAP-like solution. Its follow-up BLADE (Zheng et al., 2026) targets the full posterior by embedding the same statistical linearization in a split Gibbs sampler, alternating a diffusion prior step with a covariance-preconditioned Langevin likelihood step. KPS shares the ensemble linearization but uses it in a single closed-form Kalman update of $p(x \mid x_t)$, whose local prior covariance comes from the denoiser, rather than as guidance or inside a Langevin dynamics.

Table 1: Samplers compared in Section 5.1. Columns indicate whether each method handles non-linear forward models, needs no gradient of the forward model, needs no explicit likelihood, and targets the full posterior rather than a MAP-like solution.

Method	Posterior	Non-linear	Gradient-free	Likelihood-free
DPS (Chung et al., 2023)	✓	✓	×	×
MMPS (Rozet et al., 2024)	✓	✓	×	×
DAPS (Zhang et al., 2025)	✓	✓	×	×
PnP-DM (Wu et al., 2024)	✓	✓	×	×
EnKG (Zheng et al., 2025a)	×	✓	✓	✓
TDS (Wu et al., 2023)	✓	✓	×	×
KPS-Gradient (ours, Algorithm 1)	✓	✓	×	✓
KPS-Hybrid (ours, Algorithm 1)	✓	✓	✓	✓

4 METHOD

Our method (Figure 1) named Kalman Posterior Sampling (KPS) rests on three observations: (i) sampling a diffusion model conditionally on y reduces to sampling a single conditional distribution $p(x_{k-1} \mid x_k, y)$ at each noise level (Section 4.1), (ii) a Gaussian surrogate of this conditional can be estimated from simulated pairs (Section 4.2), and (iii) because the resulting correction acts on the clean variable only, it must be followed by a step restoring consistency with the noisy one (Section 4.3).

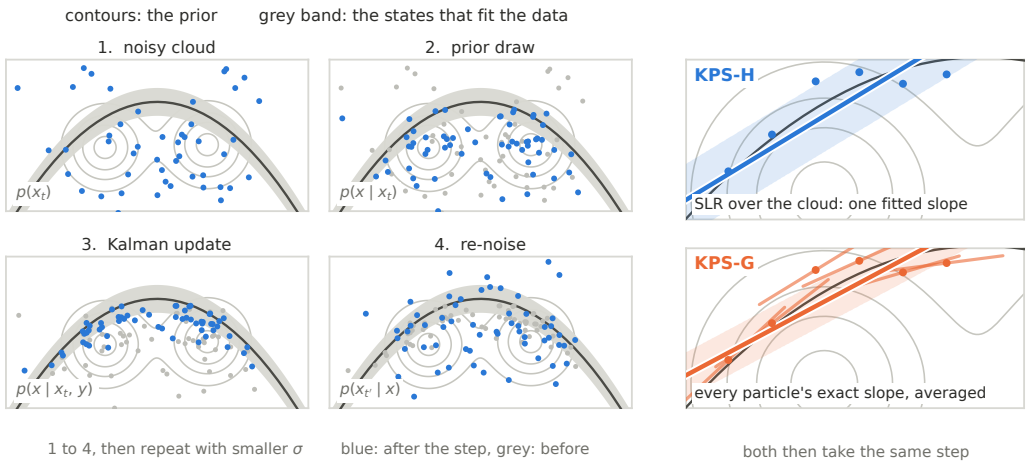

Figure 1: One noise level of KPS on a 2D example. Noisy particles are denoised into draws from $p(x \mid x_t)$, moved toward the band by the Kalman update (Equation 17), and re-noised. Right: A is fitted once over the ensemble (KPS-H) or computed from each particle's Jacobian (KPS-G).

4.1 ANNEALED POSTERIOR SAMPLING

Sampling a diffusion model amounts to reversing the forward process of Equation 4, which in practice is discretized over a schedule $0 = t_0 < t_1 < \cdots < t_T = 1$. Writing $x_k \triangleq x_{t_k}$, so that $x_0 = x$, multiplying the resulting reverse Markov chain by $p(y \mid x)/p(y)$ and telescoping the ratio along it (Appendix A.1) yields

$$p(x \mid y) \approx \int p(x_T) \prod_{k=1}^{T} p(x_{k-1} \mid x_k, y)\, \mathrm{d}x_{1:T}, \tag{10}$$

where $p(x_T) \approx \mathcal{N}(0, \Sigma_1)$ stands in for $p(x_T \mid y)$, the observation carrying almost no information at the highest noise level. Posterior sampling therefore requires only the ability to sample the transitions $p(x_{k-1} \mid x_k, y)$. Since y is conditionally independent of x_{k-1} given x, these factor as

$$p(x_{k-1} \mid x_k, y) = \int p(x \mid x_k, y)\, p(x_{k-1} \mid x_k, x)\, \mathrm{d}x, \tag{11}$$

where a bridge $p(x_{k-1} \mid x_k, x)$ can be analytically defined by the kernel in Equation 5 (Lai et al., 2025), so a draw from the transition can be obtained by sampling the local posterior and pushing it through the bridge. This composition contributes no error of its own, and the whole complexity therefore lies in approximating

$$p(x \mid x_k, y) \propto p(x \mid x_k)\, p(y \mid x), \tag{12}$$

whose two factors are exactly what the problem provides, the first through the generative model and the second through the forward model of Equation 1. Neither can be evaluated, but both can be sampled.

4.2 CORRECTING FOR THE OBSERVATION

We approximate the prior factor as

$$p(x \mid x_t) \approx \mathcal{N}(x \mid \mu_t, V_t), \tag{13}$$

a common approximation in posterior sampling methods (Rozet et al., 2024; Zhang et al., 2025), with $\mu_t = \mathbb{E}[x \mid x_t]$, and $V_t = \alpha_t^{-1}\sigma_t^2 \nabla_{x_t}^\top \mathbb{E}[x \mid x_t]$ following Tweedie's formulas as detailed in Appendix E.2. The likelihood is approximated as an affine Gaussian model

$$p(y \mid x) \approx \mathcal{N}(y \mid Ax + b, \Sigma_y). \tag{14}$$

Fitted over the whole cloud, whose particles are denoised from different noisy anchors $x_t^{(i)}$, this surrogate makes the joint of (x, y) over the cloud Gaussian with moments $(\mu_x, \mu_y, C_{xx}, C_{yx}, C_{yy})$, and identifying blocks (Appendix A.3) gives

$$A = C_{yx}C_{xx}^\dagger, \qquad b = \mu_y - A\mu_x, \qquad \Sigma_y = C_{yy} - AC_{xx}A^\top, \tag{15}$$

a construction known as statistical linearization. It assumes nothing about the noise η of Equation 1. the surrogate matches the first two moments of the joint, whatever their origin, and Σ_y absorbs both the noise of the simulator and the residual of the affine fit.

Combining this shared likelihood with the local prior of Equation 13 and conditioning on y gives, for each particle,

$$p(x \mid x_t, y) \approx \mathcal{N}(\mu_t + K(y - A\mu_t - b),\ (I - KA)V_t)\,, \qquad K = V_t A^\top (\Sigma_y + AV_tA^\top)^{-1}. \tag{16}$$

We call the resulting sampler Kalman Posterior Sampling (KPS), since the observation enters through the Kalman update 16, applied once per noise level of Equation 10, rather than through a score-based guidance term. Equation 16 is never explicitly formed. Instead we draw an ensemble $x^{(i)} \sim p(x \mid x_t^{(i)})$, simulate one observation per particle through Equation 1, fit the surrogate on the resulting pairs, and move each particle from its own position,

$$x^{(i)+} = x^{(i)} + K\left(y - y^{(i)}\right), \tag{17}$$

which admits the moments of Equation 16 (Appendix A.4). Querying the stochastic simulator directly makes the innovation carry a draw of the observation noise, so no perturbation or inflation is needed. Only draws from $p(y \mid x)$ are used, so the update applies to non-linear, non-differentiable and black-box forward models.

It remains to estimate the coefficients matrix A and the observation covariance Σ_y. KPS-Hybrid (KPS-H) takes the empirical moments of the pairs $\{x^{(i)}, y^{(i)}\}$, fitted once on the whole ensemble, each particle contributing a single pair, so A has rank at most $N-1$. KPS-Gradient (KPS-G) instead uses the Jacobian of the forward model,

$$A = \mathbb{E}[\nabla_x \mathcal{A}(x, \eta)] \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_x \mathcal{A}(x^{(i)}, \eta^{(i)}), \tag{18}$$

which by Stein's lemma are the same coefficients A whenever the ensemble is Gaussian, and which is full rank whatever the ensemble size. In practice, the average slope enters through combination of Jacobian-vector products applied per particle, this technically costs N jvp per particle. For implementation speed concern, we retain only each particle's own term, $A^{(i)} = \nabla_x \mathcal{A}(x^{(i)}, \eta^{(i)})$ as in Rozet et al. (2024). This remains an unbiased estimate of Equation 18 at one product per particle, local to that particle rather than shared across the ensemble. The observation covariance is estimated the same way in both variants and from the pairs alone. The identity in Equation 15 holds only at the least-squares A, which neither of ours is, so we take $\Sigma_y = \sigma_y^2 I$ from a two-fold held-out residual on the pairs, at no extra simulator cost (Appendix A.6). Table 1 summarizes what each variant requires.

4.3 RESTORING FORWARD CONSISTENCY

The update of Section 4.2 enforces consistency between x and the observation, but it is approximate and acts on x alone, so the pair (x, x_t) it leaves behind is no longer a plausible realization of the

forward process of Equation 4. Since Equation 10 assumes $x_t \sim p(x_t \mid y)$ at every level, this mismatch is passed to the next transition and accumulates along the chain. Rather than moving directly to the next level, we therefore alternate, at fixed t, between

$$x \sim p(x \mid x_t, y) \quad \text{and} \quad x_t \sim p(x_t \mid x, y) = p(x_t \mid x), \tag{19}$$

the second block being available in closed form from Equation 5, since x_t is conditionally independent of y given x. This is a two-block Gibbs sampler on $p(x, x_t \mid y)$ and leaves $p(x_t \mid y)$ invariant (Appendix A.7). It is the likelihood analogue of the corrector of PPR (Rochman-Sharabi & Louppe, 2026), who re-noise through the forward kernel to sample under hard constraints and show that the exact kernel has the target marginal as its unique stationary distribution. It also lets the chain move between modes, which the unimodal update Equation 16 cannot do on its own, and improves the update itself, since re-noising from a corrected x yields an x_t that already carries y, so the surrogate is fitted where the posterior has mass rather than where the prior does.

Related schemes approximate the inner conditional with Langevin dynamics (Zhang et al., 2025; Wu et al., 2024) or, under hard constraints, with a projection through the denoiser (Rochman-Sharabi & Louppe, 2026). For such approximate correctors, it is shown in PPR that the error splits into a mixing term that decays with the number of sweeps and the bias of the inner approximation. The loop therefore does not remove the bias of the update at a given level and its role is to prevent per-level errors from accumulating (Appendix A.7).

4.4 Kalman Posterior Sampling

Algorithm 1 runs Equation 10 in parallel on N particles, which, as in Zheng et al. (2025a), make the surrogate estimable and couple only through it. Each call to UPDATE costs N denoiser solves and N simulations, for $M \times T$ calls in total. The gains $K^{(i)}$ are never assembled explicitly. For KPS-H, the factored A and isotropic Σ_y reduce Equation 16 to an $r \times r$ solve, and for KPS-G the system is solved with a matrix-free GMRES implementation (Saad & Schultz, 1986), since V_t, the Jacobian of a learned denoiser, is not constrained to be symmetric nor positive definite (Appendix A.5).

Algorithm 1 Kalman Posterior Sampling

Require: y, schedule $\{t_k\}_{k=0}^{T}$, sweeps M

$x_T^{(i)} \sim \mathcal{N}(0, \sigma_T^2 I)$
for $k = T, \dots, 1$ **do**
  for $m = 1, \dots, M$ **do**
    $\{x^{(i)}\} \leftarrow \text{UPDATE}(\{x_k^{(i)}\}, y)$
    if $m < M$ **then**
      $x_k^{(i)} \sim p(x_k \mid x^{(i)})$
    else
      $x_{k-1}^{(i)} \sim p(x_{k-1} \mid x^{(i)}, x_k^{(i)})$
    end if
  end for
end for
return $\{x^{(i)}\}$

Algorithm 2 UPDATE$(\{x_t^{(i)}\}, y)$

Require: ensemble size N, rank r

$x^{(i)} \sim p(x \mid x_t^{(i)})$ ▷ Eq. 6
$y^{(i)} \leftarrow \mathcal{A}(x^{(i)}, \eta^{(i)})$ ▷ Eq. 1
$A \leftarrow$ rank-r regression, or averaged Jacobian
$\sigma_y^2 \leftarrow$ two-fold held-out residual
$K^{(i)} \leftarrow V_t^{(i)} A^\top (A V_t^{(i)} A^\top + \sigma_y^2 I)^{-1}$
$x^{(i)} \leftarrow x^{(i)} + K^{(i)}(y - y^{(i)})$
return $\{x^{(i)}\}$

5 Experiments

5.1 Latent diffusion model

To facilitate comparison with the baselines introduced in Section 3, we first consider a scientific problem for which the observation model follows the additive Gaussian formulation of Equation 2. Inspired by the 3D-Var data assimilation problem in meteorology (Lorenc et al., 2000), we consider a two-dimensional incompressible fluid governed by the Navier–Stokes equation with random forcing on the torus $\mathbb{T}^2 = [0, 2\pi]^2$ (Chen et al., 2024), and aim to reconstruct its state at a given time from a sparse and noisy observation. More precisely, the observation is obtained by averaging the 256×256

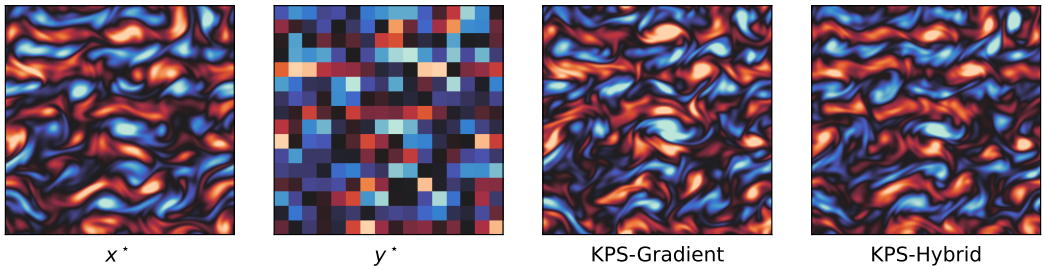

Figure 2: Latent diffusion task. A reference state $x^\star$ drawn from the latent prior and decoded, its block-averaged noisy observation $y^\star$, and one decoded posterior sample from each KPS variant.

state of the fluid over 16×16 pixel blocks and adding independent Gaussian noise with standard deviation 0.1. An example of a fluid state and its associated observation is shown in Figure 2.

To introduce a more complex and non-linear observation model, we learn the prior not directly in the pixel space, but in a compressed latent space. Following Rozet et al. (2025), we train a convolutional autoencoder with a compression factor of 32, mapping the 256×256 fluid state to a $32 \times 8 \times 8$ latent representation. The variable of interest is therefore the latent variable z and, since it must first be passed through the decoder before applying the coarsening, the operator becomes non-linear:

$$y = \text{Coarsening}\left(\text{Decoder}(z)\right) + \eta, \quad \eta \sim \mathcal{N}(0, 0.1^2 I). \tag{20}$$

Using a diffusion model trained in this latent space (Section 2), we construct 50 reference triplets (z^*, x^*, y^*) by sampling z^* according to Equation 6, decoding it into the pixel space, and applying the coarsening operator. For each reference triplet and each method in Table 1, we then generate 64 posterior samples, resulting in $50 \times 64 = 3200$ samples per method. These samples are used to compute standard metrics for posterior sampling, together with MIRA (Sharief et al., 2026), a recent score designed to compare posterior sampling methods. The resulting metrics are given in Figure 3.

Method	Skill	SSR	CRPS
DPS	0.249	0.867	285.8
MMPS	0.241	<u>0.923</u>	<u>274.9</u>
DAPS	0.312	**1.038**	357.6
PnP-DM	0.274	0.421	351.2
EnKG	0.251	0.837	288.6
TDS	0.323	0.604	396.4
KPS-G	**0.233**	1.169	**266.3**
KPS-H	<u>0.234</u>	**1.038**	**266.3**

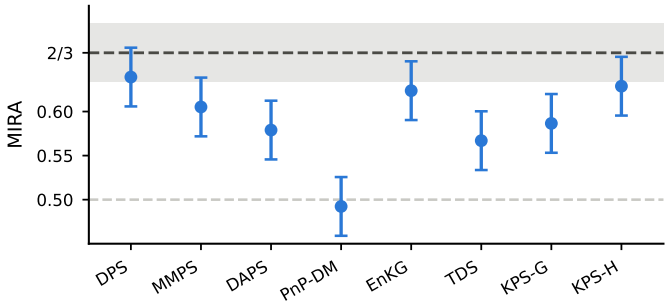

Figure 3: Skill, spread-to-skill ratio (SSR), continuous ranked probability score (CRPS) and MIRA score (Sharief et al., 2026) for the methods introduced in Section 3 on the latent diffusion problem.

Although designed for harder observation models, KPS is the best method on all three metrics in this setting (Figure 3). KPS-Gradient has the lowest skill, both variants share the lowest CRPS, and KPS-Hybrid ties with DAPS for the SSR closest to one while maintaining similar skill as the best reported. KPS-Hybrid's MIRA score is also within uncertainty of the calibrated value $2/3$ whereas KPS-Gradient is slightly over-dispersed. The samples (Figure 2 and Appendix D) show that KPS reproduces realistic vortical structures, whereas PnP-DM samples are over-smoothed, DAPS samples show high-frequency artifacts, and TDS is under-dispersed, likely from weight degeneracy in this 2048-dimensional latent space.

5.2 InverseBench

InverseBench (Zheng et al., 2025b) collects inverse problems whose forward models are the solvers used in the corresponding fields rather than synthetic degradations, which makes it a realistic testbed for the setting of Section 1.1. We use three of its tasks, each relaxing a different assumption. **Linear**

inverse scattering recovers a 128×128 permittivity map from 20 transmitters and 360 receivers, with a linear differentiable forward model and additive Gaussian noise, the setting the baselines were designed for. **Navier–Stokes** recovers a vorticity field from a later snapshot, through a non-linear PDE solver whose gradients exist but require differentiating a full trajectory, which is unaffordable here. **Black-hole imaging** recovers a 64×64 image from interferometric visibilities, non-linear and with noise that is non-Gaussian in image space. See Zheng et al. (2025b) for the specifications and Appendix B.1 for our configurations.

Table 2: Point accuracy and calibration on InverseBench for each task. We report benchmark metrics along with additional calibration metrics for our methods. Baselines are the two best published results, obtained with an explicit likelihood: RED-diff and DDNM (inverse scattering), DPS and PnP-DM (black-hole imaging), EnKG at $N = 2048$ and DPG (Navier-Stokes). The full tables can be found in Appendix B.2.

	Inverse scattering			Black-hole imaging			Navier–Stokes		
Method	PSNR ↑	CRPS ↓	spread-skill → 1	blur PSNR ↑	CRPS ↓	spread-skill → 1	rel. L_2 ↓	CRPS ↓	spread-skill → 1
Best published	36.6	–	–	32.9	–	–	**0.120**	–	–
2nd published	36.4	–	–	32.9	–	–	0.325	–	–
KPS-G (N=16)	**39.5**	0.008	0.80	36.3	0.141	0.96	–	–	–
KPS-H (N=512)	21.6	0.095	0.53	**36.9**	0.147	0.64	0.313	0.199	0.25

KPS-Gradient exceeds the best published point accuracy on both tasks where gradient is available (Table 2). KPS-Hybrid does so on black-hole imaging, is on par with DPG on Navier–Stokes, and trails on inverse scattering. These comparisons are indicative only, since the baselines evaluate an explicit likelihood (with a different noise model on black-hole imaging) while KPS only queries the simulator. More importantly, a reconstruction scored against one truth says nothing about whether the remaining uncertainty is right, and Sharief et al. (2026) find that methods doing well on InverseBench point-wise accuracy are not calibrated.

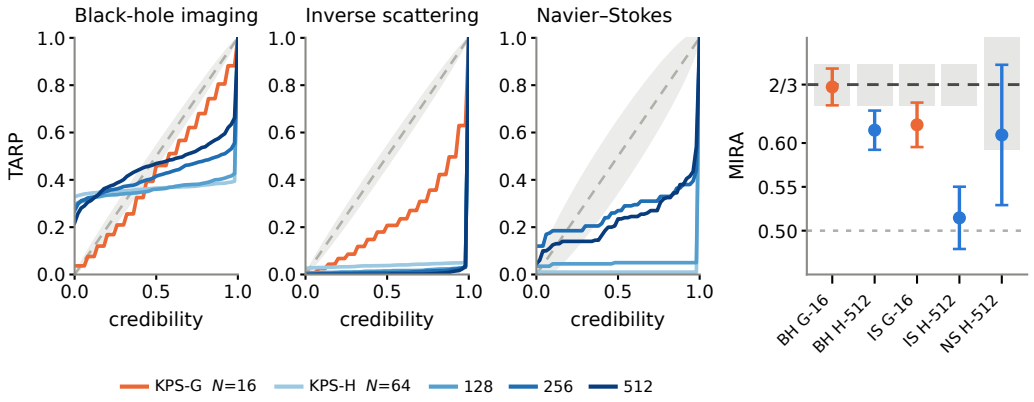

Figure 4: Calibration on InverseBench. Left: TARP is the expected coverage against credibility level. The diagonal is exact calibration, curves below it are over-confident. Right: MIRA, where $2/3$ is calibrated and $1/2$ uninformative. BH, IS, and NS are the three tasks. G-16 is KPS-Gradient at $N = 16$, H-512 is KPS-Hybrid at $N = 512$. Navier-Stokes is Hybrid only, its gradients being unaffordable.

We therefore score in Figure 4 our posterior estimation with TARP and MIRA (Lemos et al., 2023; Sharief et al., 2026), which scale to these dimensions and need no tractable likelihood. Appendix B.3 adds spread, skill, CRPS and coverage, which are easier to read but weaker. KPS-Gradient is the better calibrated variant, close to calibrated on black-hole imaging and mildly over-confident on inverse scattering, while KPS-Hybrid is under-dispersed and improves with N.

Aggregate scores also hide the shape of the posterior. Black-hole imaging is ill-posed enough that distinct images reproduce the same visibilities, and Figure 5 shows two such modes from one run. A sampler concentrating on either would score well on point accuracy while generating the wrong set of possible states. Neither accuracy nor calibration asks whether the samples, pushed back through the simulator, reproduce the observation they were conditioned on. This check needs no ground truth and costs one additional simulation per sample. In Appendix B.4, KPS-Gradient is close to calibrated in observation space at $N = 16$ and KPS-Hybrid approaches it as N grows. We suggest this evaluation as a default for posterior sampling rather than a diagnostic specific to KPS.

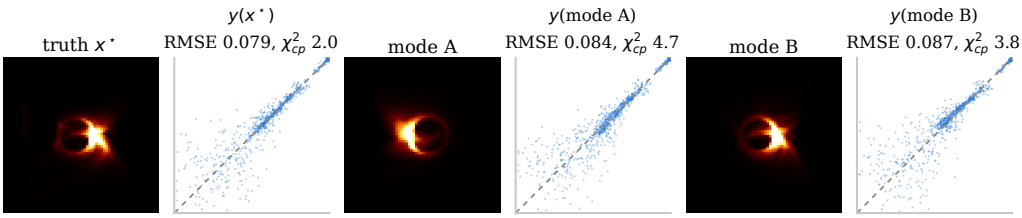

Figure 5: Two distinct images from one KPS-Hybrid ensemble on black-hole imaging, next to the truth. Each scatter plot compares visibilities re-simulated from the state to its left with the true conditioning observation. Both modes fit the data with an error close to that of the truth, so y alone cannot separate them.

6 Discussion

Choosing a variant. The two variants trade simulator access for ensemble size. When differentiating through the simulator is affordable, KPS-Gradient is the safer choice. With only $N = 16$ particles, it gave our best point accuracy and calibration, at the cost of derivatives that dominate wall time. Otherwise, as for Navier-Stokes, KPS-Hybrid needs only simulations, but its low-rank surrogate calls for hundreds of particles, remains under-dispersed at the tested sizes, and occasionally diverges (Table 8). Simulators with cheaply differentiable parts could combine both linearizations.

Scoring the posterior. Accuracy against a single truth rewards samplers that collapse onto one plausible solution (Sharief et al., 2026). We therefore advocate posterior level scores such as TARP and MIRA, together with the posterior predictive check of Appendix B.4, which needs no ground truth. With a single pair $(x^\star, y^\star)$ per instance, however, calibration is only assessed across instances, never against the posterior of a given observation. Benchmarks with reference posteriors (Yao et al., 2026; Zach et al., 2025) make this conditional check possible and are a natural next testbed.

Limitations and extensions. The Gibbs loop keeps per-level errors from accumulating but does not remove the bias of the Gaussian surrogate, which grows with the non-linearity of the simulator and likely contributes to the under-dispersion of KPS-Hybrid and the closure-phase misfit on black-hole imaging. The ensemble Kalman literature offers remedies such as localization and inflation for rank-deficient ensembles (Houtekamer & Mitchell, 2001; Anderson & Anderson, 1999), iterated updates that relinearize at a fixed noise level (Bocquet & Sakov, 2014), and Gaussian mixtures for multimodal local posteriors. Drawing $p(x \mid x_t)$ with stochastic flow maps (Holderrieth et al., 2026; Potaptchik et al., 2026; McCallum et al., 2026) would also reduce the cost of each update.

Conclusion. Scientific forward models are stochastic simulators rather than closed-form likelihoods. KPS couples a diffusion prior to such simulators through a Kalman update that needs only simulations and gradients when available, and yields accurate posteriors on realistic inverse problems, close to calibrated when gradients are affordable. We hope that scoring the posterior, not only the reconstruction, becomes standard practice for simulator-based posterior sampling.

AI USE STATEMENT

In this work, we did not use generative AI tools for any of the tasks for which the ICLR 2027 policy requires disclosure. In particular, we have not used them to develop the theoretical framework, formulate mathematical claims or write their proofs, propose hypotheses, design the method or experiments, implement methods, generate synthetic datasets (the reference data of Section 5.1 are sampled from diffusion models we trained ourselves), clean or reformat datasets, or interpret results, and translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure: to write and adjust the plotting code used to render figures, and to edit the paper for readability, namely shortening passages, rephrasing, suggesting structure, and correcting grammar. These tools were not used to produce new scientific content or to develop any part of the method. We have reviewed all AI-assisted work: the authors checked every edited passage to ensure that it preserves the intended meaning and technical content, and all figures display results from our own experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. URL http://jmlr.org/papers/v26/23-1605.html.

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: https://doi.org/10.1016/0304-4149(82)90051-5. URL https://www.sciencedirect.com/science/article/pii/0304414982900515.

Jeffrey L. Anderson and Stephen L. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741 – 2758, 1999. doi: 10.1175/1520-0493(1999)127⟨2741:AMCIOT⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/127/12/1520-0493_1999_127_2741_amciot_2.0.co_2.xml.

Johann Baumeister. *Ill-posed problems*, pp. 16–26. Vieweg+Teubner Verlag, Wiesbaden, 1987. ISBN 978-3-322-83967-1. doi: 10.1007/978-3-322-83967-1_2. URL https://doi.org/10.1007/978-3-322-83967-1_2.

M. Bocquet and P. Sakov. An iterative ensemble kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, 140(682):1521–1535, 2014. doi: https://doi.org/10.1002/qj.2236. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.2236.

M. Bouillon, S. Safieddine, S. Whitburn, L. Clarisse, F. Aires, V. Pellet, O. Lezeaux, N. A. Scott, M. Doutriaux-Boucher, and C. Clerbaux. Time evolution of temperature profiles retrieved from 13 years of infrared atmospheric sounding interferometer (iasi) data using an artificial neural network. *Atmospheric Measurement Techniques*, 15(6):1779–1793, 2022. doi: 10.5194/amt-15-1779-2022. URL https://amt.copernicus.org/articles/15/1779/2022/.

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael Samuel Albergo, Nicholas Matthew Boffi, and Eric Vanden-Eijnden. Probabilistic forecasting with stochastic interpolants and föllmer processes. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=UQYXZdca92.

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.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117. URL https://www.pnas.org/doi/abs/10.1073/pnas.1912789117.

Masoumeh Dashti and Andrew M. Stuart. *The Bayesian Approach to Inverse Problems*, pp. 311–428. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-12385-1_7. URL https://doi.org/10.1007/978-3-319-12385-1_7.

Michael Deistler, Jan Boelts, Peter Steinbach, Guy Moss, Thomas Moreau, Manuel Gloeckler, Pedro L. C. Rodrigues, Julia Linhart, Janne K. Lappalainen, Benjamin Kurt Miller, Pedro J. Gonçalves, Jan-Matthis Lueckmann, Cornelius Schröder, and Jakob H. Macke. Simulation-Based Inference: A Practical Guide, 2025.

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 12069–12095, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/333bbde5ed6397cd9bc5bac552b3b7a6-Paper-Conference.pdf.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. 2008. URL https://api.semanticscholar.org/CorpusID:122725027.

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.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tAtpSwjsCB.

P. L. Houtekamer and Herschel L. Mitchell. A sequential ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123 – 137, 2001. doi: 10.1175/1520-0493(2001)129⟨0123:ASEKFF⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/129/1/1520-0493_2001_129_0123_asekff_2.0.co_2.xml.

Marco A Iglesias, Kody J H Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, March 2013. ISSN 1361-6420. doi: 10.1088/0266-5611/29/4/045001. URL http://dx.doi.org/10.1088/0266-5611/29/4/045001.

Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

Pablo Lemos, Adam Coogan, Yashar Hezaveh, and Laurence Perreault-Levasseur. Sampling-based accuracy testing of posterior estimators for general inference, 2023. URL https://arxiv.org/abs/2302.03026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

A. C. Lorenc, S. P. Ballard, R. S. Bell, N. B. Ingleby, P. L. F. Andrews, D. M. Barker, J. R. Bray, A. M. Clayton, T. Dalby, D. Li, T. J. Payne, and F. W. Saunders. The met. office global three-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 126(570):2991–3012, 2000. doi: https://doi.org/10.1002/qj.49712657002. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49712657002.

Yosra Marnissi, Yuling Zheng, Emilie Chouzenoux, and Jean-Christophe Pesquet. A variational bayesian approach for image restoration - application to image deblurring with poisson-gaussian noise. *IEEE Trans. Computational Imaging*, 3(4):722–737, 2017. doi: 10.1109/TCI.2017.2700203. URL https://doi.org/10.1109/TCI.2017.2700203.

Sam McCallum, Zander W. Blasingame, Timothy Herschell, Niklas Rindtorff, Alexander Tong, and James Foster. Strong stochastic flow maps, 2026. URL https://arxiv.org/abs/2606.01086.

Ewan Pinnington, Peter Lean, Mihai Alexe, Eulalie Boucher, Simon Lang, Patrick Laloyaux, Gert Mertes, Tomas Kral, Patricia de Rosnay, Matthew Chantry, and Anthony McNally. Aifs-dop: End-to-end medium-range weather prediction from observations alone with machine learning, 2026. URL https://arxiv.org/abs/2606.19093.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael Samuel Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=K5gV8Yptne.

Omer Rochman-Sharabi and Gilles Louppe. Predict-project-renoise: Sampling diffusion models under hard constraints, 2026. URL https://arxiv.org/abs/2601.21033.

François Rozet, Gérôme Andry, Francois Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=7v88Fh6iSM.

François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, Francois Lanusse, and Shirley Ho. Lost in latent space: An empirical study of latent diffusion models for physics emulation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=xoNrbfbekM.

Youcef Saad and Martin H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, July 1986. ISSN 0196-5204. doi: 10.1137/0907058. URL http://dx.doi.org/10.1137/0907058.

Sammy Nasser Sharief, Justine Zeghal, Gabriel Missael Barco, Pablo Lemos, Yashar Hezaveh, and Laurence Perreault-Levasseur. MIRA: A score for conditional distribution accuracy and model comparison. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ra2t1V4nml.

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

M. C. K. Tweedie. Functions of a statistical variate with given means, with special reference to laplacian distributions. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43 (1):41–49, 1947. doi: 10.1017/S0305004100023185.

Luhuan Wu, Brian L. Trippe, Christian A Naesseth, John Patrick Cunningham, and David Blei. Practical and asymptotically exact conditional sampling in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=eWKqr1zcRv.

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

Tongda Xu, Xiyan Cai, Xinjie Zhang, Xingtong Ge, Dailan He, Ming Sun, Jingjing Liu, Ya-Qin Zhang, Jian Li, and Yan Wang. Rethinking diffusion posterior sampling: From conditional score estimator to maximizing a posterior. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GcvLoqOoXL.

Jiachen Yao, Zi-Siang Hsu, Xi Deng, Aditi Gupta, Xin Ju, Sally M Benson, Gege Wen, and Anima Anandkumar. Posteriorbench: From point estimates to posterior matching in evaluating generative inverse solvers, 2026. URL https://arxiv.org/abs/2609.20794.

Martin Zach, Youssef Haouchat, and Michael Unser. A statistical benchmark for diffusion posterior sampling algorithms, 2025. URL https://arxiv.org/abs/2509.12821.

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

Qinsheng Zhang and Yongxin Chen. Fast Sampling of Diffusion Models with Exponential Integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

Hongkai Zheng, Wenda Chu, Austin Wang, Nikola Borislavov Kovachki, Ricardo Baptista, and Yisong Yue. Ensemble kalman diffusion guidance: A derivative-free method for inverse problems, 2025a. URL https://openreview.net/forum?id=ykt6I21YQZ.

Hongkai Zheng, Wenda Chu, Bingliang Zhang, Zihui Wu, Austin Wang, Berthy Feng, Caifeng Zou, Yu Sun, Nikola Borislavov Kovachki, Zachary E Ross, Katherine Bouman, and Yisong Yue. Inversebench: Benchmarking plug-and-play diffusion models for scientific inverse problems. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=U3PBITXNG6.

Hongkai Zheng, Austin Wang, Zihui Wu, Zhengyu Huang, Ricardo Baptista, and Yisong Yue. Blade: A derivative-free bayesian inversion method using diffusion priors, 2026. URL https://arxiv.org/abs/2510.10968.

A THE KPS UPDATE IN DETAIL

A.1 ANNEALED CONDITIONAL CHAIN

Restricted to the schedule $0 = t_0 < \cdots < t_T = 1$, the forward process of Equation 4 is a Markov chain $x_0 \to x_1 \to \cdots \to x_T$ with $x_0 = x$. Its reversal is Markov as well,

$$p(x_{0:T}) = p(x_T) \prod_{k=1}^{T} p(x_{k-1} \mid x_k). \tag{21}$$

The observation depends on the path only through x_0, so conditioning on y multiplies Equation 21 by the single ratio $p(y \mid x_0)/p(y)$, which telescopes along the chain,

$$\frac{p(y \mid x_0)}{p(y)} = \frac{p(y \mid x_T)}{p(y)} \prod_{k=1}^{T} \frac{p(y \mid x_{k-1})}{p(y \mid x_k)}. \tag{22}$$

Pairing each factor of Equation 22 with the corresponding factor of Equation 21,

$$p(x_{0:T} \mid y) = \underbrace{p(x_T)\,\frac{p(y \mid x_T)}{p(y)}}_{= p(x_T \mid y)} \prod_{k=1}^{T} \underbrace{p(x_{k-1} \mid x_k)\,\frac{p(y \mid x_{k-1})}{p(y \mid x_k)}}_{= p(x_{k-1} \mid x_k,\, y)}. \tag{23}$$

The first identification is Bayes' rule. The second is Bayes' rule at fixed x_k, using $p(y \mid x_{k-1}, x_k) = p(y \mid x_{k-1})$ as x_k is independent of x_0, hence of y, given x_{k-1}. Integrating out $x_{1:T}$ gives the exact identity

$$p(x \mid y) = \int p(x_T \mid y) \prod_{k=1}^{T} p(x_{k-1} \mid x_k, y)\, \mathrm{d}x_{1:T}. \tag{24}$$

Equation 10 replaces $p(x_T \mid y)$ by $p(x_T) \approx \mathcal{N}(0, \Sigma_1)$. This is its only approximation, and it is mild, since $p(x_T \mid y)/p(x_T) = p(y \mid x_T)/p(y) \to 1$ as the noise at $t_T = 1$ erases what x_T carries about x.

Bridge. Marginalizing each transition over the clean variable,

$$p(x_{k-1} \mid x_k, y) = \int p(x \mid x_k, y)\, p(x_{k-1} \mid x_k, x, y)\, \mathrm{d}x, \tag{25}$$

where $p(x_{k-1} \mid x_k, x, y) = p(x_{k-1} \mid x_k, x)$ since the noisy variables are independent of y given x, which gives Equation 11. For $s < t$, the forward transition is $p(x_t \mid x_s) = \mathcal{N}(\alpha_{t|s} x_s, \sigma_{t|s}^2 I)$ with $\alpha_{t|s} = \alpha_t/\alpha_s$ and $\sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2$. Bayes' rule on this kernel and on $p(x_s \mid x)$ of Equation 5 gives a valid bridge in closed form (as developed in Lai et al. (2025)),

$$p(x_s \mid x_t, x) = \mathcal{N}\left(\frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\, x_t + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2}\, x,\ \frac{\sigma_s^2 \sigma_{t|s}^2}{\sigma_t^2}\, I \right), \tag{26}$$

used with $s = t_{k-1}$ and $t = t_k$ in line 8 of Algorithm 1.

A.2 PRIOR-FACTOR MOMENTS

The mean and covariance of Equation 13 both come from the denoiser. Rearranging Equation 7 gives $\mathbb{E}[x \mid x_t] = \alpha_t^{-1}\big(x_t + \sigma_t^2 \nabla_{x_t} \log p(x_t)\big)$, and differentiating it with respect to x_t,

$$\nabla_{x_t}^{\top} \mathbb{E}[x \mid x_t] = \alpha_t^{-1}\big(I + \sigma_t^2 \nabla_{x_t}^2 \log p(x_t)\big). \tag{27}$$

The second-order Tweedie identity for the kernel of Equation 5 is $\mathbb{V}[x \mid x_t] = \alpha_t^{-2}\sigma_t^2\big(I + \sigma_t^2 \nabla_{x_t}^2 \log p(x_t)\big)$, so combining the two eliminates the Hessian and leaves

$$V_t \triangleq \mathbb{V}[x \mid x_t] = \alpha_t^{-1}\sigma_t^2\, \nabla_{x_t}^{\top} \mathbb{E}[x \mid x_t], \tag{28}$$

which is the expression used in Section 4.2. Only a vector–Jacobian product through the denoiser is ever needed, never the matrix itself. Note that V_t is the covariance of $p(x \mid x_t)$, not of the cloud. The cloud covariance C_{xx} enters only the fit of the slope (Appendix A.3). Substituting it for V_t in the gain would use the wrong prior factor in Equation 12.

A.3 GAUSSIAN SURROGATE

Two Gaussians enter the update, with different roles. The likelihood surrogate is fitted once over the whole cloud, whose particles are denoised from different anchors, so that its law at level t is

$$\pi_t(x) = \int p(x \mid x_t)\, p(x_t \mid y)\, \mathrm{d}x_t. \tag{29}$$

It is then combined, particle by particle, with the local prior $\mathcal{N}(\mu_t^{(i)}, V_t^{(i)})$ of Equation 13. At high noise, $\pi_t \approx p(x)$; as the anchors carry y, it approaches $p(x \mid y)$, so the linearization is taken where the posterior has mass (Section 4.3).

Fitting over the cloud. Let $x \sim \pi_t$ and $y \sim p(y \mid x)$, and approximate their joint by the Gaussian with the same first two moments,

$$p(x, y) \approx \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} \right), \qquad C_{xy} = C_{yx}^\top. \tag{30}$$

If $x \sim \mathcal{N}(\mu_x, C_{xx})$ and $y = Ax + b + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \Sigma_y)$ independent of x, the joint must take the form

$$\mathcal{N}\left(\begin{bmatrix} \mu_x \\ A\mu_x + b \end{bmatrix}, \begin{bmatrix} C_{xx} & C_{xx}A^\top \\ AC_{xx} & AC_{xx}A^\top + \Sigma_y \end{bmatrix} \right), \tag{31}$$

where $\mathrm{Cov}(y, x) = \mathbb{E}\big[(A(x - \mu_x) + \varepsilon)(x - \mu_x)^\top\big] = AC_{xx}$, and the lower block follows from the independence of ε and x. Equating Equation 31 and Equation 30 block by block,

$$C_{yx} = AC_{xx}, \qquad \mu_y = A\mu_x + b, \qquad C_{yy} = AC_{xx}A^\top + \Sigma_y, \tag{32}$$

and solving in order gives Equation 15,

$$A = C_{yx}C_{xx}^{-1}, \qquad b = \mu_y - A\mu_x, \qquad \Sigma_y = C_{yy} - AC_{xx}A^\top = C_{yy} - C_{yx}C_{xx}^{-1}C_{xy}. \tag{33}$$

KPS-Hybrid uses the empirical moments of the pairs $\{x^{(i)}, y^{(i)}\}$. Nothing requires η to be additive or Gaussian: only the moments of (x, y) enter, and Σ_y absorbs both the stochasticity of $\mathcal{A}$ and the misfit of the affine model.

Updating each particle. The fitted likelihood is shared across particles but the prior is not. For each particle, dropping the index i, the local prior $\mathcal{N}(\mu_t, V_t)$ and the fitted likelihood give the joint

$$p(x, y \mid x_t) \approx \mathcal{N}\left(\begin{bmatrix} \mu_t \\ A\mu_t + b \end{bmatrix}, \begin{bmatrix} V_t & V_tA^\top \\ AV_t & AV_tA^\top + \Sigma_y \end{bmatrix} \right), \tag{34}$$

whose conditional on y is Equation 16, with $K = V_tA^\top(AV_tA^\top + \Sigma_y)^{-1}$. The cloud covariance C_{xx} enters only the fit of A.

Least-squares view. As explained in Zheng et al. (2025a), the same (A, b) minimize $\mathbb{E}_{\pi_t}\|y - Ax - b\|^2$, whose stationarity conditions are $b = \mu_y - A\mu_x$ and $AC_{xx} = C_{yx}$. For an arbitrary slope, the residual covariance is

$$\mathrm{Cov}(y - Ax) = C_{yy} - AC_{xy} - C_{yx}A^\top + AC_{xx}A^\top, \tag{35}$$

which reduces to $C_{yy} - AC_{xx}A^\top$ only when $AC_{xx} = C_{yx}$. Neither variant uses that slope, since KPS-Hybrid truncates its rank and KPS-Gradient uses Jacobians, so Σ_y is estimated as a residual covariance directly (Appendix A.6).

Rank. Estimated from N pairs, C_{xx} has rank at most $N - 1 \ll n$, hence the pseudo-inverse $A = C_{yx}C_{xx}^\dagger$: the minimum-norm least-squares slope, which predicts y along the directions spanned by the centred cloud and is zero on their complement.

A.4 ENSEMBLE REALIZATION

Drawing the ensemble. Each particle is denoised from its own anchor by integrating Equation 6 from t down to 0, costing a few denoiser evaluations and no simulator calls. The draw must be a *sample* of $p(x \mid x_t)$ rather than the Tweedie mean as the surrogate of Equation 15 is estimated from the scatter of the pairs $\{x^{(i)}, y^{(i)}\}$, and a cloud collapsed onto the conditional mean carries no scatter to fit. In practice, we use a deterministic probability-flow as inner sampler rather than a stochastic reversal.

Moment matching. Equation 17 reproduces the moments of Equation 16 particle by particle, conditionally on the anchor. Given $x_t^{(i)}$, $x^{(i)}$ has mean $\mu_t^{(i)}$ and covariance $V_t^{(i)}$, and the fitted surrogate $y^{(i)} = Ax^{(i)} + b + \varepsilon^{(i)}$ with $\varepsilon^{(i)} \sim \mathcal{N}(0, \Sigma_y)$ independent of $x^{(i)}$. Dropping the index,

$$\mathbb{E}\big[x^+ \mid x_t\big] = \mu_t + K(y - A\mu_t - b), \tag{36}$$

$$\mathrm{Cov}\big[x^+ \mid x_t\big] = (I - KA)V_t(I - KA)^\top + K\Sigma_y K^\top = (I - KA)V_t, \tag{37}$$

the last equality holding for the gain of Equation 16. These moments are over the draw of the particle and of the simulator noise, not over the ensemble, which enters only through (A, b, Σ_y). Each updated particle is thus one draw from its own local posterior, the first block of Equation 19, and the updated cloud is a mixture of such draws over anchors rather than an ensemble approximating a single Gaussian. The cross term vanishes because $\varepsilon^{(i)}$ is independent of $x^{(i)}$: querying the simulator with its noise makes the innovation $y - y^{(i)}$ carry a draw of $\mathcal{N}(0, \Sigma_y)$, which removes the perturbed observations a stochastic ensemble Kalman filter would need.

Incremental versus anchored updates. Each particle moves from its own position. The anchored alternative $x^{(i)+} = \mu_t + K(y - y^{(i)})$ places every particle at a common point plus a residual-driven offset, so its analysis covariance is $K\Sigma_y K^\top$ alone: the prior spread V_t is discarded rather than contracted, and the ensemble collapses. The distinction is a property of the update *form*, not of the quality of its ingredients. The anchored form collapses even with an exact A and an exact Σ_y, whereas Equation 17 retains $(I - KA)V_t$, the posterior covariance of the surrogate.

A.5 Low-rank estimation and linear solves

KPS-Hybrid. The fitted slope is kept factored, $A = WV_h^\top$ with r columns, never as a $d \times n$ matrix. With $M = V_h^\top V_t V_h$, $G = W^\top W$ and $\Sigma_y = \sigma_y^2 I$, the push-through identity

$$W^\top\big(WMW^\top + \sigma_y^2 I\big)^{-1} = \big(\sigma_y^2 I_r + GM\big)^{-1} W^\top \tag{38}$$

reduces the whole step to one $r \times r$ solve. The equivalent Woodbury form is algebraically identical but numerically fragile here: it recovers the step as a difference divided by σ_y^2, and since $AV_tA^\top$ has rank r while d is in the thousands, $\sigma_y^2 I$ is the only term making the system invertible and that difference is a near-exact cancellation over the remaining $d - r$ directions, so dividing it by a small σ_y^2 amplifies the rounding error by $1/\sigma_y^2$. Equation 38 never divides by σ_y^2. We additionally project M onto the positive semi-definite cone, which is required rather than cosmetic. V_t of Equation 28 is the Jacobian of a learned network and is not symmetric, so M can be indefinite, and an exact solve of an indefinite system returns an amplifying step rather than a gain.

KPS-Gradient. Here A is implicit and admits no low-rank factorization, so the inverse is applied by GMRES (Saad & Schultz, 1986) on $u \mapsto AV_tA^\top u + \sigma_y^2 u$, each iteration costing one Jacobian-vector and one vector-Jacobian product through the simulator and one through the denoiser, following the same idea as Rozet et al. (2024). GMRES rather than conjugate gradients because the operator is not guaranteed symmetric nor positive definite. For an exact denoiser V_t would be proportional to a Hessian by Equation 28, but a learned network's Jacobian is the Hessian of nothing. CG assumes symmetry to keep its search directions conjugate and is unstable here, whereas GMRES assumes neither symmetry nor definiteness.

A.6 Estimating Σ_y out of sample

For a given slope, the covariance the surrogate needs is that of the residual, Equation 35. Its in-sample estimate is biased low since a regression on N pairs has up to $N - 1$ free directions. At full rank it interpolates its own pairs and returns a zero residual however poor the linearization is. We therefore use two-fold cross-validation. Split the particles into halves $\mathcal{I}_1, \mathcal{I}_2$, fit (A_f, b_f) on $\mathcal{I}_f$ with the regression of Appendix A.5, and score each fit on the other half,

$$\sigma_y^2 = \frac{1}{Nd} \sum_{f=1}^{2} \sum_{i \in \mathcal{I}_{3-f}} \big\| y^{(i)} - A_f x^{(i)} - b_f \big\|_2^2, \qquad \Sigma_y = \sigma_y^2 I. \tag{39}$$

Both folds reuse the simulated pairs, so the estimate costs no simulator call. A full $d \times d$ covariance cannot be estimated from $N \ll d$ pairs. An isotropic one can, and it keeps the solve of Appendix A.5 at size $r \times r$. Because σ_y^2 contains both the simulator noise and the misfit of the affine model, it shrinks the gain $K = V_t A^\top (A V_t A^\top + \sigma_y^2 I)^{-1}$ exactly where the linearization is poor, which prevents over-stepping at low noise levels, where the misfit is large relative to V_t.

A.7 Gibbs loop at fixed noise level

At fixed t, one sweep of Equation 19 draws the clean variable and re-noises it through the forward kernel, which defines the Markov kernel

$$K_t(x_t' \mid x_t) = \int p(x_t' \mid x)\, p(x \mid x_t, y)\, \mathrm{d}x, \qquad p(x \mid x_t, y) \propto p(x \mid x_t)\, p(y \mid x). \tag{40}$$

This is the corrector of Rochman-Sharabi & Louppe (2026), introduced to sample under a hard constraint $x \in \mathcal{C}$, with the likelihood $p(y \mid x)$ in place of the indicator $\mathbf{1}_{\mathcal{C}}(x)$ in the inner conditional.

Invariance and convergence. If $x_t \sim p(x_t \mid y)$,

$$\int p(x' \mid x_t, y)\, p(x_t \mid y)\, \mathrm{d}x_t = p(x' \mid y), \tag{41}$$

$$\int p(x_t' \mid x')\, p(x' \mid y)\, \mathrm{d}x' = \int p(x_t' \mid x', y)\, p(x' \mid y)\, \mathrm{d}x' = p(x_t' \mid y), \tag{42}$$

using $p(x_t \mid x, y) = p(x_t \mid x)$, so $p(x_t \mid y)$, the marginal Equation 10 assumes at every level, is stationary for K_t. Since $p(x_t' \mid x)$ is Gaussian with full support, K_t has a density that is positive everywhere, which makes this stationary distribution unique and gives $K_t^M q \to p(x_t \mid y)$ as $M \to \infty$ for any initial law q, the likelihood analogue of Rochman-Sharabi & Louppe (2026, Proposition 3.1).

Approximate corrector. KPS replaces $p(x \mid x_t, y)$ by the surrogate $q(x \mid x_t, y)$ of Equation 16, giving $\hat{K}_t(x_t' \mid x_t) = \int p(x_t' \mid x)\, q(x \mid x_t, y)\, \mathrm{d}x$, whose stationary distribution is no longer $p(x_t \mid y)$. Rochman-Sharabi & Louppe (2026, Appendix C) decompose the error of such an approximate corrector, with a point estimate as inner approximation, into a mixing term that decays exponentially with M and an irreducible bias of the inner approximation. In KPS the second term is the bias of the Gaussian surrogate. The sweeps forget the initialization of the level, not the bias of q.

Consistency and contraction. Let r be the law of the cloud after an update, and $(rP_t)(x_t) = \int r(x)\, p(x_t \mid x)\, \mathrm{d}x$ its push-forward through Equation 5. After the re-noise, the pair has law $r(x)\, p(x_t \mid x)$, whose conditional is the forward kernel exactly, whatever the error in r. As $p(x_t \mid y) = \big(p(\cdot \mid y)P_t\big)(x_t)$, both laws go through the same kernel, and the data-processing inequality gives, for any f-divergence D,

$$D\big(rP_t \,\|\, p(x_t \mid y)\big) \le D\big(r \,\|\, p(x \mid y)\big). \tag{43}$$

The re-noise therefore never amplifies the error it receives and, since P_t adds noise, typically contracts it. This is a single-step statement, not a bound on the whole chain. Without it, the uncorrected x_k would enter x_{k-1} directly through the bridge of Equation 26. As noted by Rochman-Sharabi & Louppe (2026), re-noising also restores the noise level and injects the stochasticity the corrector needs to mix. Here it lets the next update, refitted around particles that already carry y, reach modes that the unimodal Equation 16 cannot.

Advancing the chain. As in the predictor-corrector structure of Rochman-Sharabi & Louppe (2026), each level is left by a predictor step: the last sweep replaces the re-noise by the bridge $x_{k-1} \sim p(x_{k-1} \mid x_k, x)$ (Algorithm 1, line 8). Composed with the update, this is Equation 11, so the correction of level k is carried to level $k-1$.

A.8 Variants and cost

The variants differ only in the slope A and share the prior draw, V_t, held-out Σ_y, incremental update and Gibbs loop.

Table 3: Per-particle cost of one call to UPDATE. n_{ODE} is the number of ODE steps of the prior draw, n_{K} the Krylov dimension corresponding to solver steps, and r the regression rank.

	KPS-Hybrid	KPS-Gradient
Slope A	regression on $\{x^{(i)}, y^{(i)}\}$	$\nabla_x \mathcal{A}(x^{(i)}, \eta^{(i)})$
Rank of A	$\min(r, N-1)$	full
Simulator	black box	differentiable
Linear solve	$r \times r$, Equation 38	GMRES, matrix-free
Simulator calls	1	1
Simulator JVPs / VJPs	–	n_{K} / $n_{\mathrm{K}}+1$
Denoiser evaluations (prior draw)	n_{ODE}	n_{ODE}
Denoiser VJPs (applying V_t)	r	$n_{\mathrm{K}}+1$

The simulator budget is MTN calls for both variants. Neither variant is free of differentiation. KPS-Hybrid needs no simulator derivative, but applies V_t of Equation 28 to the r columns of V_h through r vector-Jacobian products with the denoiser, after which the gain requires no further product (Appendix A.5). KPS-Gradient needs, per Krylov iteration, one simulator JVP, one simulator VJP and one denoiser VJP, plus one of each VJP to apply the final gain. These derivatives do not count against the simulator budget but dominate wall time, and n_{K} is the setting that most affects the accuracy of KPS-Gradient. Using the per-particle $A^{(i)}$ rather than the ensemble average of Equation 18 divides the simulator JVPs by N, and all products are batched over the cloud. With $A^{(i)}$, the particles of KPS-Gradient couple only through σ_y^2.

B INVERSEBENCH EXPERIMENTS

B.1 SETUP

Table 4: InverseBench problems, with forward models and priors from Zheng et al. (2025b). σ_{noise} is the additive observation noise at evaluation (black-hole imaging uses the thermal noise of the visibilities instead); the last column gives the test instances.

Problem	Forward model	σ_{noise}	Prior	Instances
Black-hole imaging	EHT visibilities, 64×64 image, `vis_thermal` noise, `direct` transform	0	`blackhole-50k`	100 (0–99)
Inverse scattering	128×128 permittivity, 20 transmitters × 360 receivers, wavenumber 6	0.0001	`inv-scatter-5m`	100 (0–99)
Navier–Stokes	vorticity at $t = 1.0$, Re $= 200.0$, 128^2 downsampled ×2	0.0001	`ns-5m`	10 (0–9)

Table 5: KPS settings on InverseBench: ensemble size N, noise levels T, Gibbs sweeps per level M, regression rank r (Hybrid), GMRES Krylov dimension (Gradient) corresponding to number of solver steps, ODE steps to draw each particle from $p(x \mid x_t)$, and noise range $[\sigma_{\min}, \sigma_{\max}]$ of the schedule.

Variant	Slope	N	T	M	r	Krylov	ODE	$\sigma_{\max}$	$\sigma_{\min}$
KPS-G	averaged Jacobian	16	32	2	–	8	4	80	0.05
KPS-H	ensemble regression	64	32	2	32	–	4	80	0.05
KPS-H	ensemble regression	128	32	2	32	–	4	80	0.05
KPS-H	ensemble regression	256	32	2	32	–	4	80	0.05
KPS-H	ensemble regression	512	32	2	32	–	4	80	0.05

B.2 POINT-WISE ACCURACY

Baseline numbers are reproduced from Zheng et al. (2025b) and were obtained with an explicit likelihood. KPS uses only simulator queries, so the comparison is indicative rather than controlled.

Table 6: Linear inverse scattering, point accuracy. Baselines as reported by Zheng et al. (2025b). KPS mean $\pm$ standard deviation over 100 instances.

Method	N	PSNR ↑	SSIM ↑
published baselines (InverseBench, Table 3, 360 receivers)			
FISTA-TV	–	32.13	0.9790
DDRM	–	32.60	0.9290
DDNM	–	36.38	0.9350
ΠGDM	–	27.93	0.8890
DPS	–	32.06	0.8460
LGD	–	27.90	0.8120
DiffPIR	–	34.24	<u>0.9880</u>
PnP-DM	–	33.91	<u>0.9880</u>
DAPS	–	34.64	0.9570
RED-diff	–	<u>36.56</u>	0.9810
FPS	–	33.24	0.8700
MCG-diff	–	30.94	0.7510
this work			
KPS-G	16	$\mathbf{39.52 \pm 2.90}$	$\mathbf{0.9962 \pm 0.0023}$
KPS-H	64	20.53 ± 4.49	0.7649 ± 0.1382
KPS-H	128	21.38 ± 4.92	0.7846 ± 0.1388
KPS-H	256	21.80 ± 5.41	0.7753 ± 0.1592
KPS-H	512	21.64 ± 5.81	0.7559 ± 0.1663

Table 7: Navier–Stokes, relative L_2 error, mean $\pm$ standard deviation. Baselines as reported by Zheng et al. (2025b). KPS over 10 instances.

Method	N	rel. L_2 ↓
published baselines (InverseBench, Table 8, $\times 2$, $\sigma = 0$)		
EKI	–	0.5770 ± 0.1380
DPS-fGSG	–	1.6870 ± 0.1560
DPS-cGSG	–	2.2030 ± 0.3140
DPG	–	0.3250 ± 0.1880
SCG	–	0.9080 ± 0.6000
EnKG	2048	$\mathbf{0.1200 \pm 0.0850}$
this work		
KPS-H	64	0.6020 ± 0.2214
KPS-H	128	0.4628 ± 0.2555
KPS-H	256	0.3717 ± 0.2626
KPS-H	512	<u>0.3132 ± 0.1836</u>

Table 8: Black-hole imaging. χ^2_{cp} and χ^2_{camp} measure the fit to closure phases and closure amplitudes (target 1). The last column counts instances with a diverged ensemble. Baselines as reported by Zheng et al. (2025b), whose noise model differs from ours.

Method	N	PSNR ↑	blur PSNR f=10 ↑	$\chi^2_{\text{cp}} \to 1$	$\chi^2_{\text{camp}} \to 1$	failures
published baselines (InverseBench, Table 6, 100% observation time)						
SMILI	–	22.67 ± 3.13	27.79 ± 4.02	1.88 ± 0.95	17.61 ± 10.30	–
EHT-Imaging	–	24.28 ± 3.63	28.57 ± 4.52	$\mathbf{1.25 \pm 0.25}$	1.26 ± 0.32	–
DPS	–	25.86 ± 3.90	32.94 ± 6.19	8.76 ± 37.78	5.46 ± 24.18	–
LGD	–	21.22 ± 3.64	26.06 ± 4.98	13.24 ± 17.23	13.23 ± 39.11	–
RED-diff	–	23.77 ± 4.13	29.13 ± 6.22	1.85 ± 0.94	2.05 ± 2.36	–
PnP-DM	–	26.07 ± 3.70	32.88 ± 6.02	1.31 ± 0.20	$\underline{1.20 \pm 0.22}$	–
DAPS	–	25.60 ± 3.64	32.78 ± 5.68	$\underline{1.30 \pm 0.32}$	1.23 ± 0.53	–
DiffPIR	–	25.01 ± 4.64	31.86 ± 6.56	3.27 ± 1.62	2.97 ± 1.20	–
this work						
KPS-G	16	$\underline{29.41 \pm 3.57}$	$\underline{36.28 \pm 4.70}$	8.28 ± 11.38	$\mathbf{1.10 \pm 0.13}$	–
KPS-H	64	28.45 ± 4.32	34.53 ± 6.08	16.33 ± 29.47	1.65 ± 0.64	1/100
KPS-H	128	29.17 ± 4.16	35.62 ± 5.71	14.01 ± 28.02	1.34 ± 0.33	4/100
KPS-H	256	29.37 ± 4.44	36.08 ± 6.28	11.13 ± 23.52	1.45 ± 1.74	4/100
KPS-H	512	$\mathbf{29.57 \pm 4.61}$	$\mathbf{36.89 \pm 6.62}$	10.87 ± 32.11	1.39 ± 1.27	–

B.3 CALIBRATION OF THE POSTERIOR

Table 9: Posterior metrics, median is reported over test instances with associated [interquartile range] to filter out outliers encountered in diverging runs. Particle RMSE (mean error of individual particles), skill (error of the ensemble mean) and CRPS measure accuracy. Spread-skill ratio (target 1) and 90% coverage (target 0.9) measure calibration.

Method	N	particle RMSE ↓	skill ↓	CRPS ↓	spread-skill ratio → 1	90% coverage → 0.9
Black-hole imaging						
KPS-G	16	0.728 [0.567, 0.851]	0.533 [0.441, 0.669]	0.1323 [0.1102, 0.1601]	0.940 [0.825, 1.067]	0.804 [0.706, 0.885]
KPS-H	64	0.348 [0.271, 0.644]	0.348 [0.270, 0.643]	0.1315 [0.0969, 0.2572]	0.068 [0.039, 0.085]	0.198 [0.131, 0.296]
KPS-H	128	0.323 [0.251, 0.504]	0.320 [0.249, 0.503]	0.1111 [0.0870, 0.1924]	0.118 [0.082, 0.146]	0.316 [0.223, 0.475]
KPS-H	256	0.332 [0.265, 0.537]	0.325 [0.258, 0.531]	0.1100 [0.0754, 0.1842]	0.206 [0.146, 0.275]	0.554 [0.402, 0.690]
KPS-H	512	0.339 [0.261, 0.574]	0.327 [0.236, 0.535]	0.1018 [0.0688, 0.1805]	0.361 [0.236, 0.448]	0.672 [0.539, 0.781]
Inverse scattering						
KPS-G	16	0.035 [0.032, 0.038]	0.028 [0.025, 0.031]	0.0083 [0.0063, 0.0101]	0.791 [0.766, 0.826]	0.892 [0.857, 0.930]
KPS-H	64	0.348 [0.276, 0.388]	0.347 [0.275, 0.387]	0.1479 [0.0936, 0.1850]	0.054 [0.051, 0.062]	0.051 [0.030, 0.547]
KPS-H	128	0.320 [0.247, 0.363]	0.319 [0.246, 0.361]	0.1336 [0.0826, 0.1686]	0.111 [0.099, 0.136]	0.078 [0.060, 0.508]
KPS-H	256	0.306 [0.224, 0.365]	0.298 [0.217, 0.345]	0.1149 [0.0680, 0.1438]	0.273 [0.230, 0.378]	0.171 [0.111, 0.698]
KPS-H	512	0.332 [0.237, 0.381]	0.289 [0.215, 0.328]	0.1029 [0.0617, 0.1326]	0.502 [0.446, 0.609]	0.312 [0.199, 0.887]
Navier–Stokes						
KPS-H	64	0.484 [0.426, 0.810]	0.484 [0.426, 0.810]	0.3440 [0.2929, 0.6181]	0.042 [0.025, 0.048]	0.068 [0.035, 0.080]
KPS-H	128	0.362 [0.300, 0.579]	0.361 [0.299, 0.578]	0.2404 [0.1971, 0.4077]	0.092 [0.064, 0.114]	0.147 [0.100, 0.179]
KPS-H	256	0.274 [0.194, 0.531]	0.269 [0.190, 0.528]	0.1784 [0.1202, 0.3518]	0.217 [0.178, 0.258]	0.283 [0.244, 0.312]
KPS-H	512	0.245 [0.184, 0.440]	0.239 [0.177, 0.431]	0.1496 [0.1071, 0.2738]	0.245 [0.207, 0.284]	0.330 [0.279, 0.371]

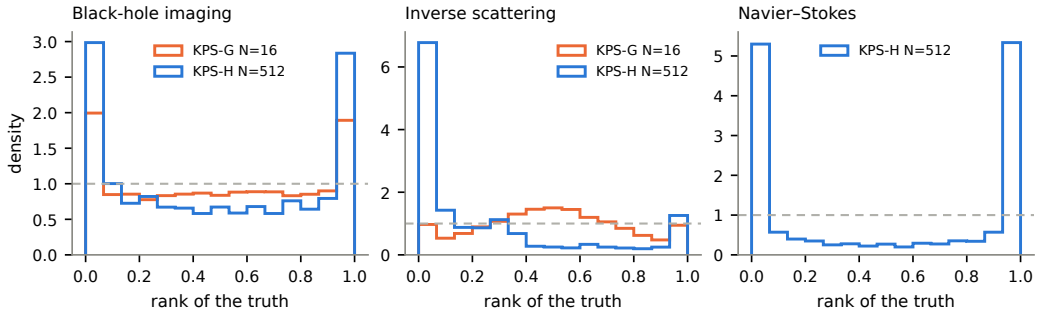

Figure 6: Rank of the truth within the ensemble, pooled over pixels and instances. A flat histogram (dashed) is calibrated. The U shape of KPS-Hybrid indicates under-dispersion.

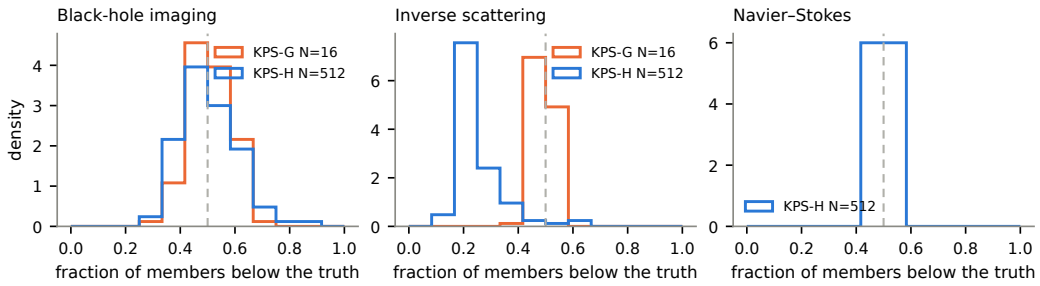

Figure 7: Fraction of ensemble particles below the truth. An unbiased posterior concentrates at 0.5 (dashed). A shift indicates bias, which the rank histogram of Figure 6 mixes with dispersion.

B.4 Calibration of the posterior predictive

Pushing each member back through the simulator gives $y^{(i)} = \mathcal{A}(x^{(i)}, \eta^{(i)})$ and turns the same diagnostics into a check in observation space, at one simulation per sample. This asks whether the ensemble reproduces the observation it was conditioned on, which is weaker than calibration in x but is the part a scientist can verify without the truth.

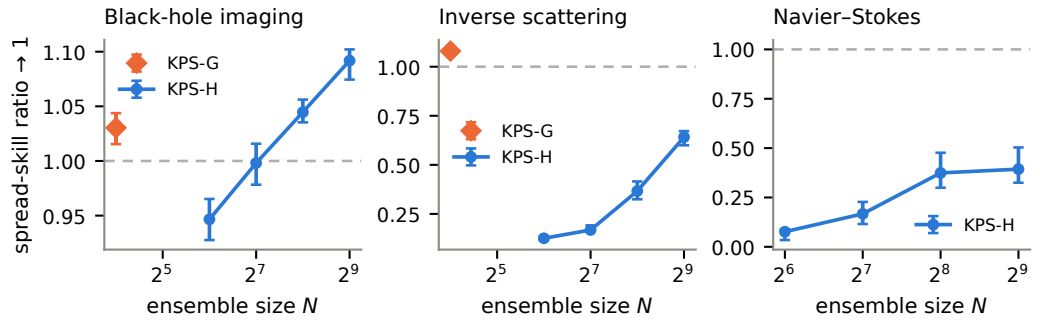

Figure 8: Spread-skill ratio of the posterior predictive $y^{(i)} = \mathcal{A}(x^{(i)}, \eta^{(i)})$ against ensemble size. 1 (dashed) is calibrated. KPS-Gradient is close to calibrated at $N = 16$. KPS-Hybrid improves with N and slightly overshoots on black-hole imaging. Compare the same ratio on x in Table 9.

C GALLERIES

C.1 EFFECT OF THE ENSEMBLE SIZE

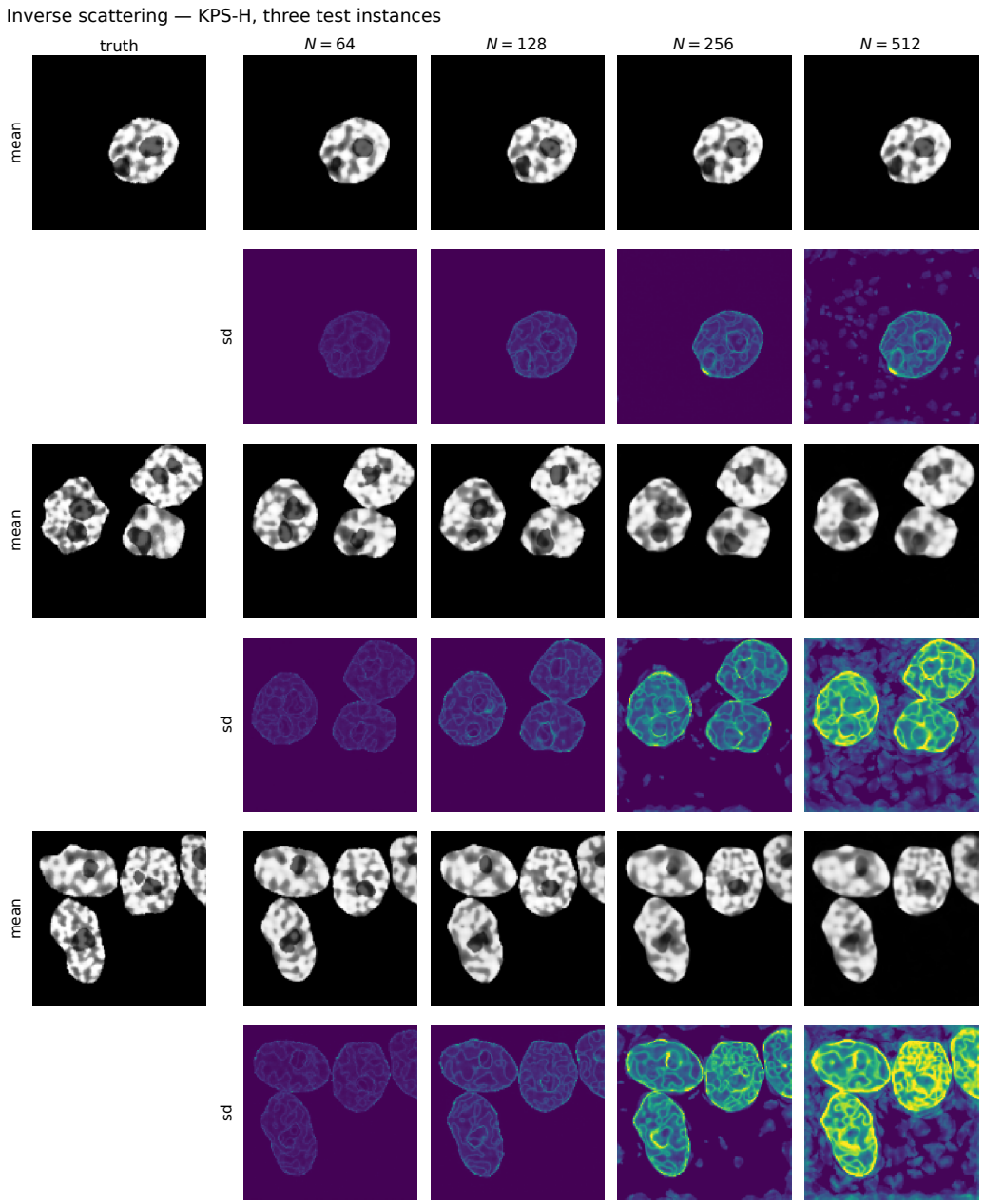

Figure 9: Effect of the ensemble size on KPS-Hybrid, linear inverse scattering. For three test instances: truth, then posterior mean (top) and standard deviation (bottom) for $N = 64$ to 512. The mean barely changes while the spread grows with N.

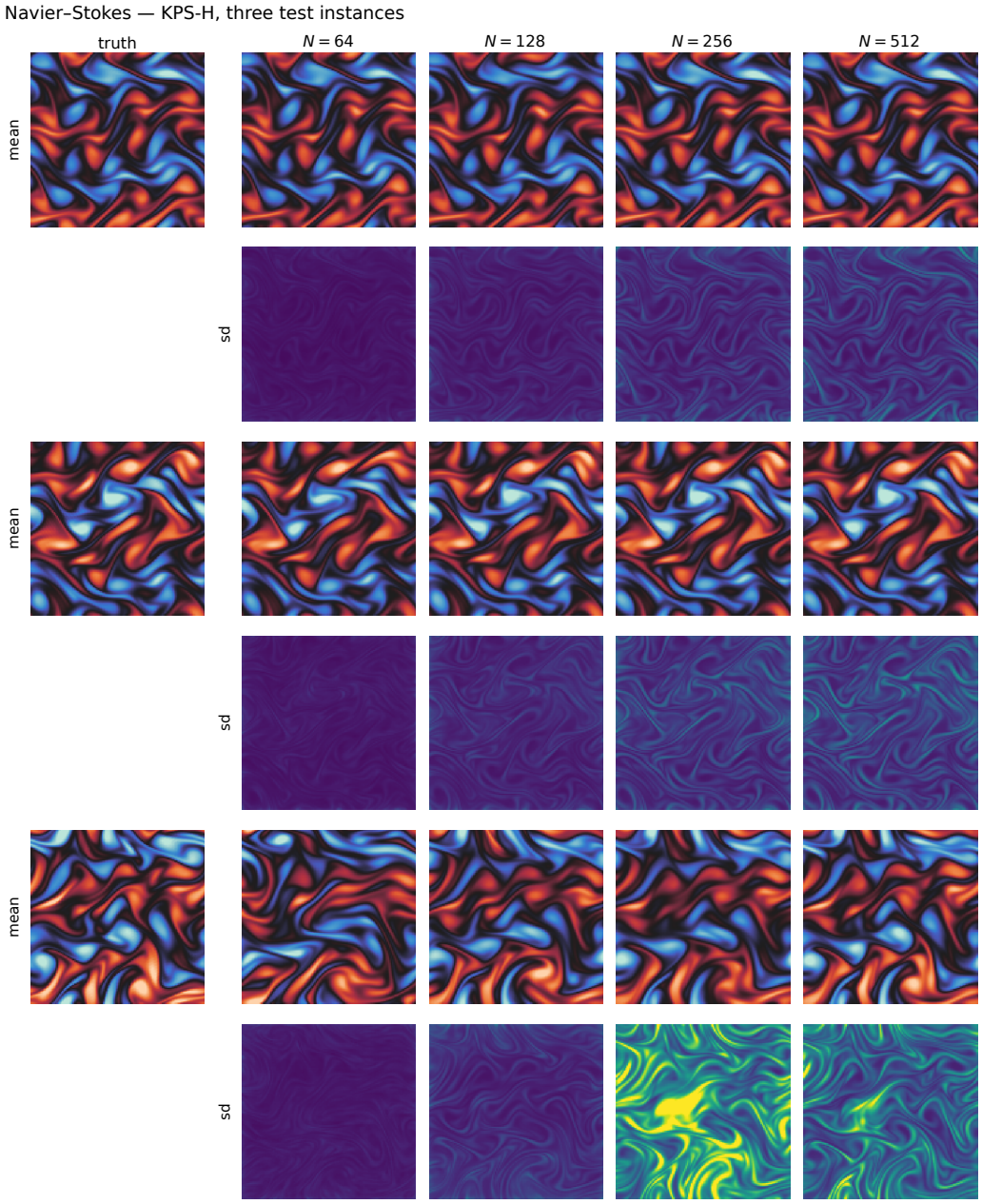

Figure 10: As Figure 9, for Navier-Stokes.

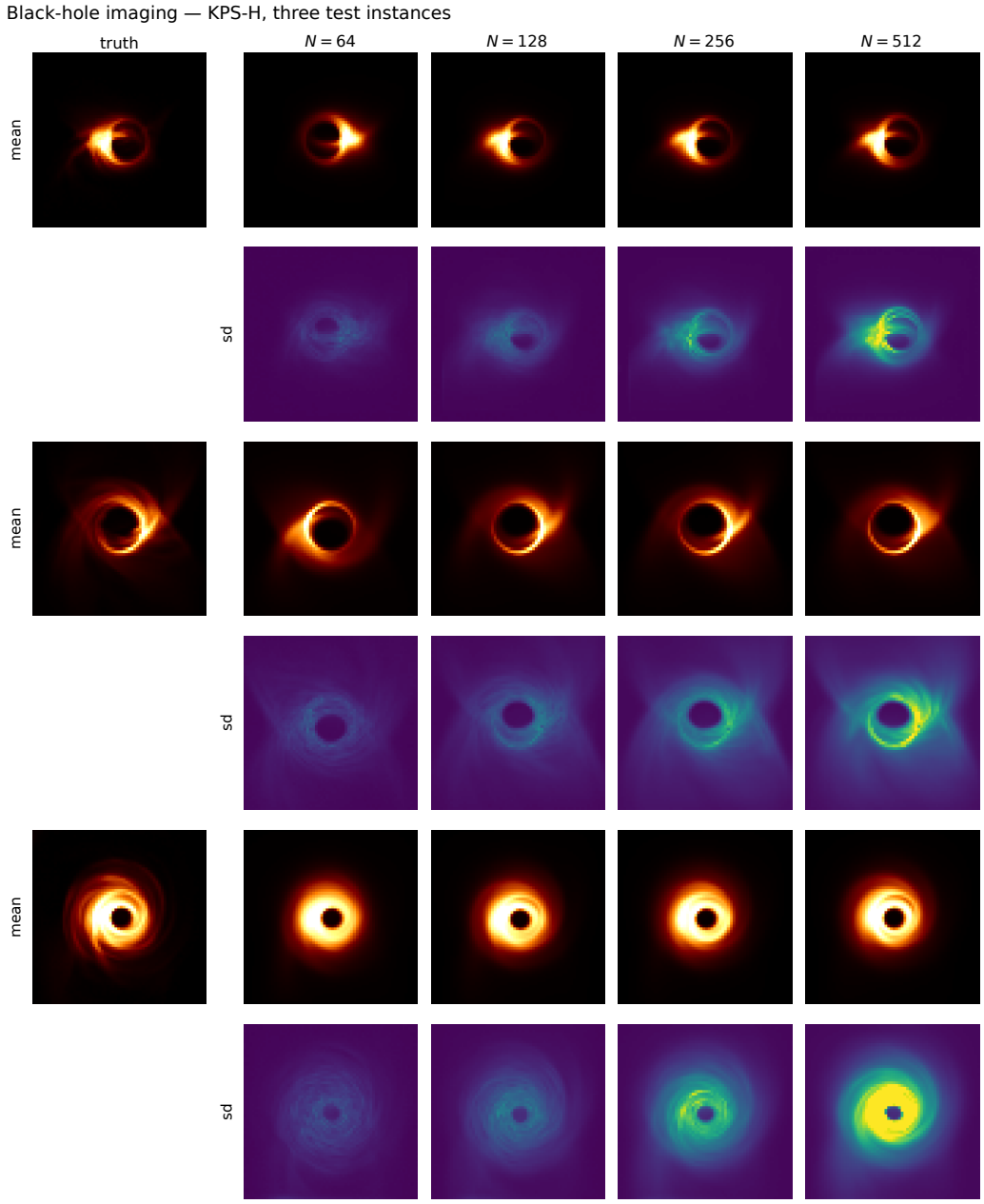

Figure 11: As Figure 9, for black-hole imaging.

C.2 POSTERIOR SAMPLES

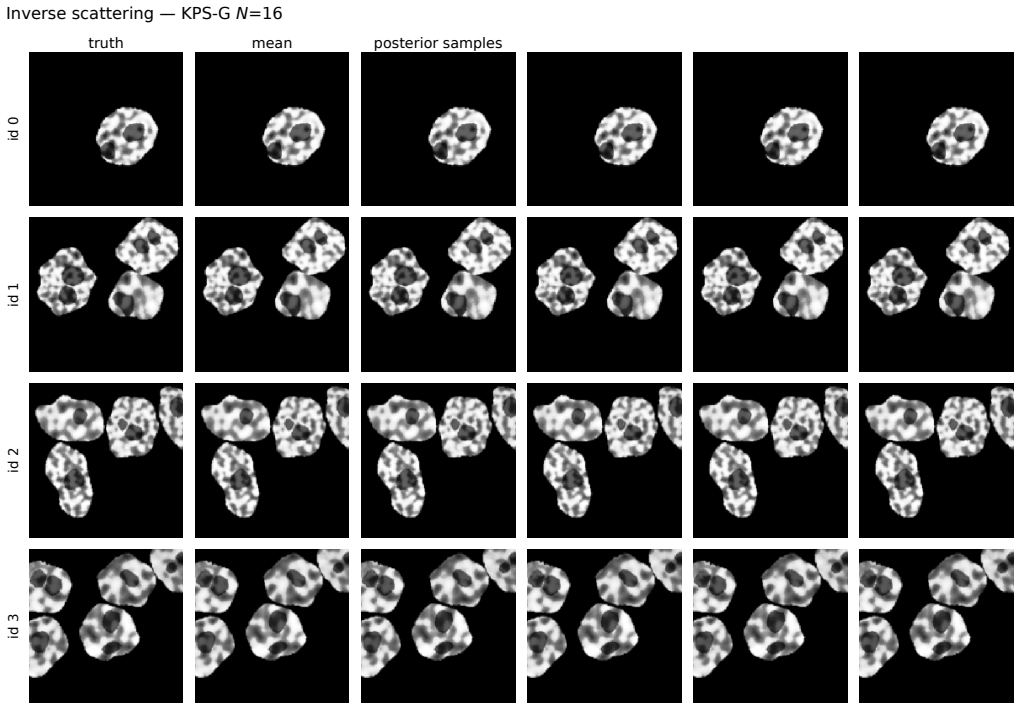

Figure 12: Posterior samples of KPS-Gradient ($N = 16$) on linear inverse scattering. Each row is a test instance: truth, posterior mean, then four ensemble members.

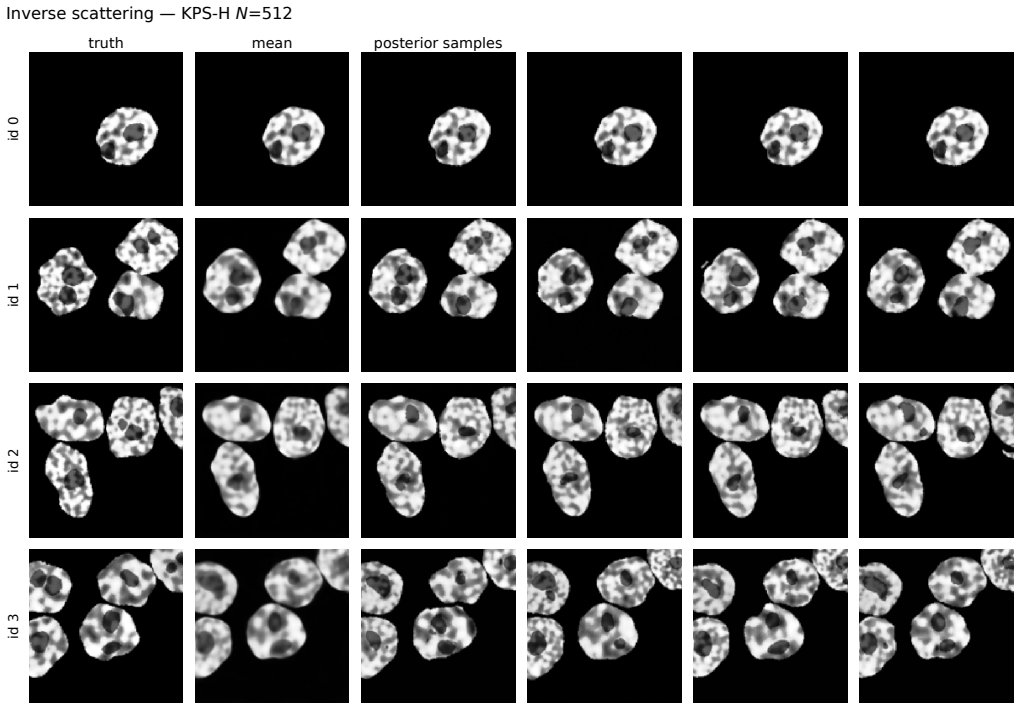

Figure 13: As Figure 12, KPS-Hybrid ($N = 512$).

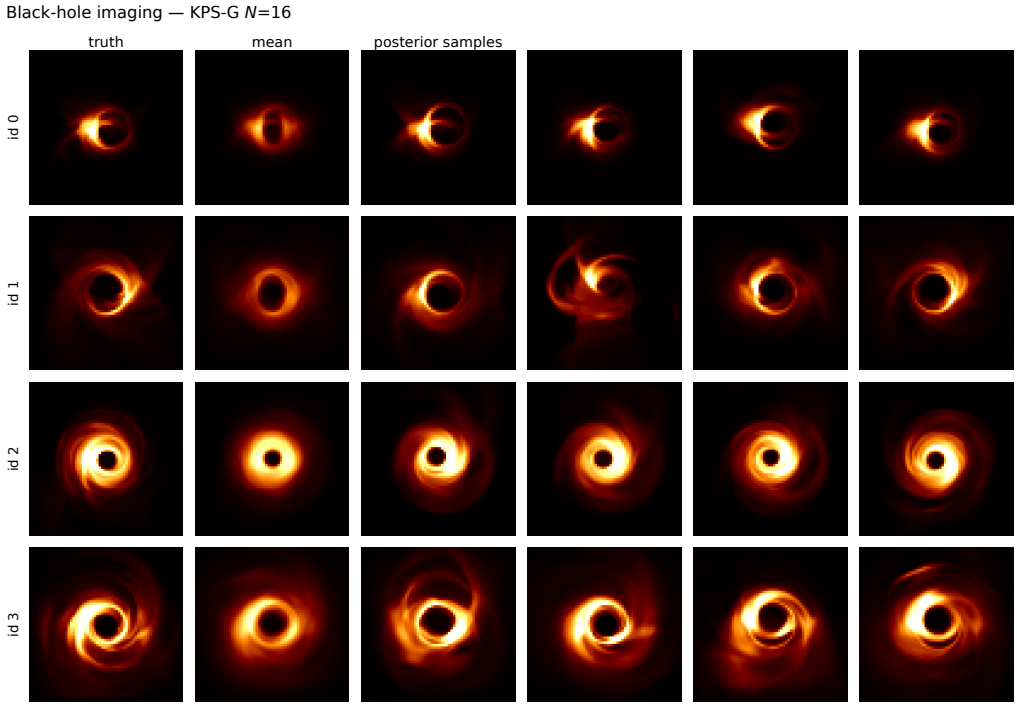

Figure 14: As Figure 12, black-hole imaging, KPS-Gradient ($N = 16$).

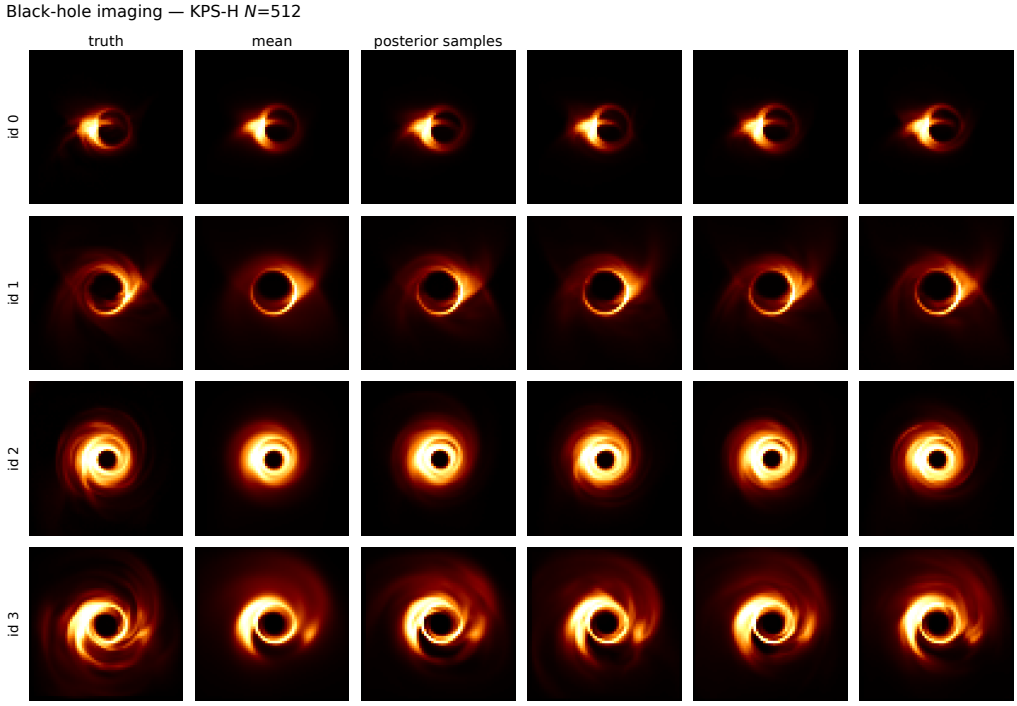

Figure 15: As Figure 12, black-hole imaging, KPS-Hybrid ($N = 512$).

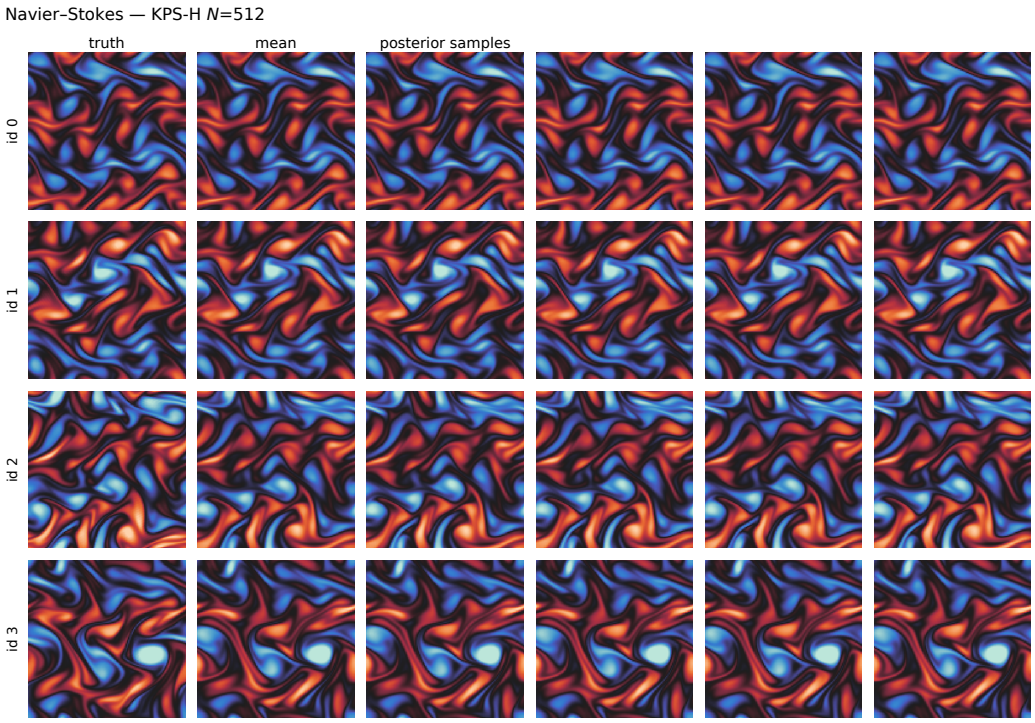

Figure 16: As Figure 12, Navier–Stokes, KPS-Hybrid ($N = 512$).

D LATENT DIFFUSION GALLERIES

Each row is one method, showing the ground truth $x^\star$, the observation $y^\star$, and three posterior samples. One figure per test instance.

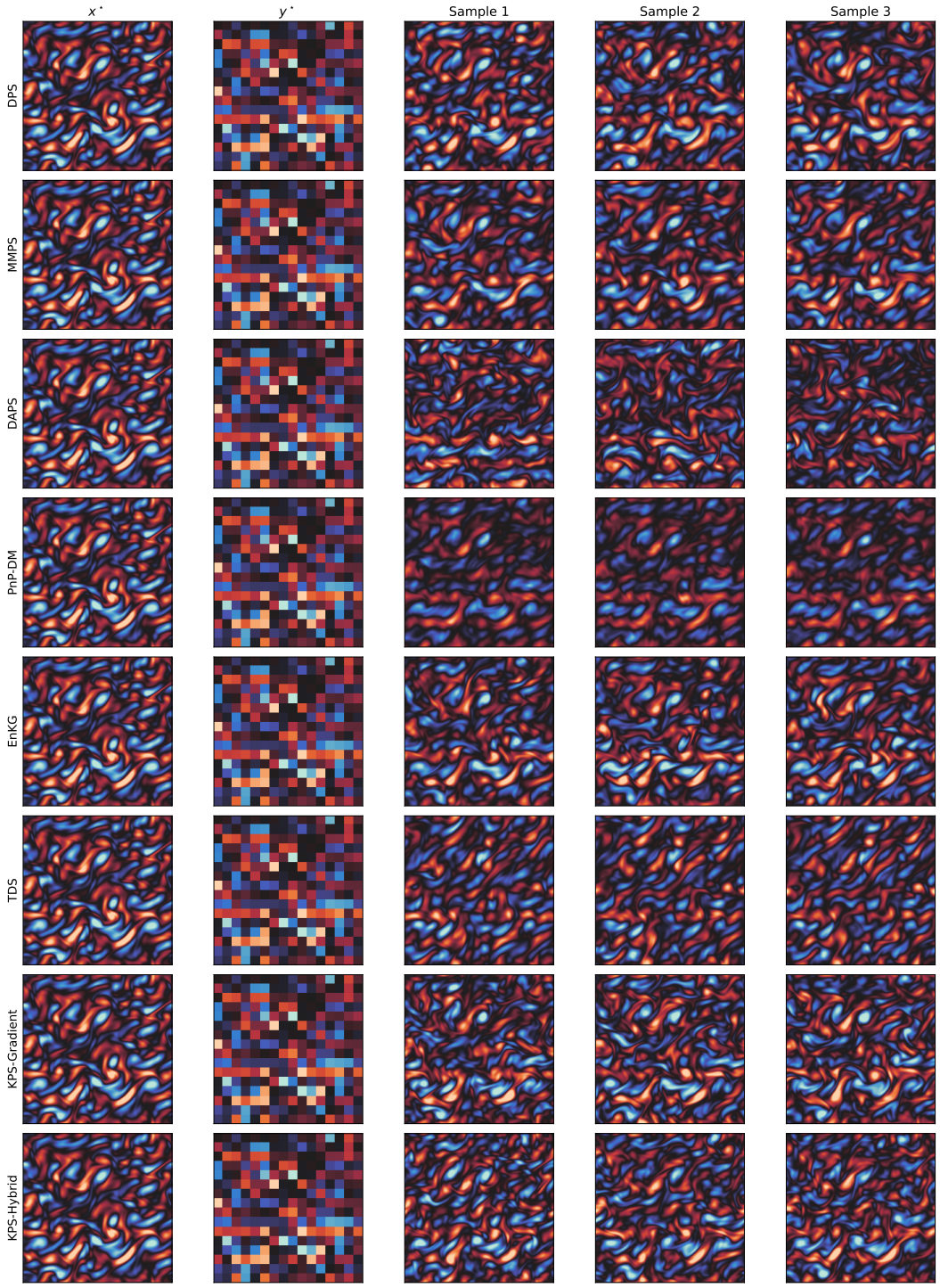

Figure 17: All methods on the latent diffusion task, test instance 3. Each row is a method: truth $x^\star$, observation $y^\star$, then three decoded posterior samples.

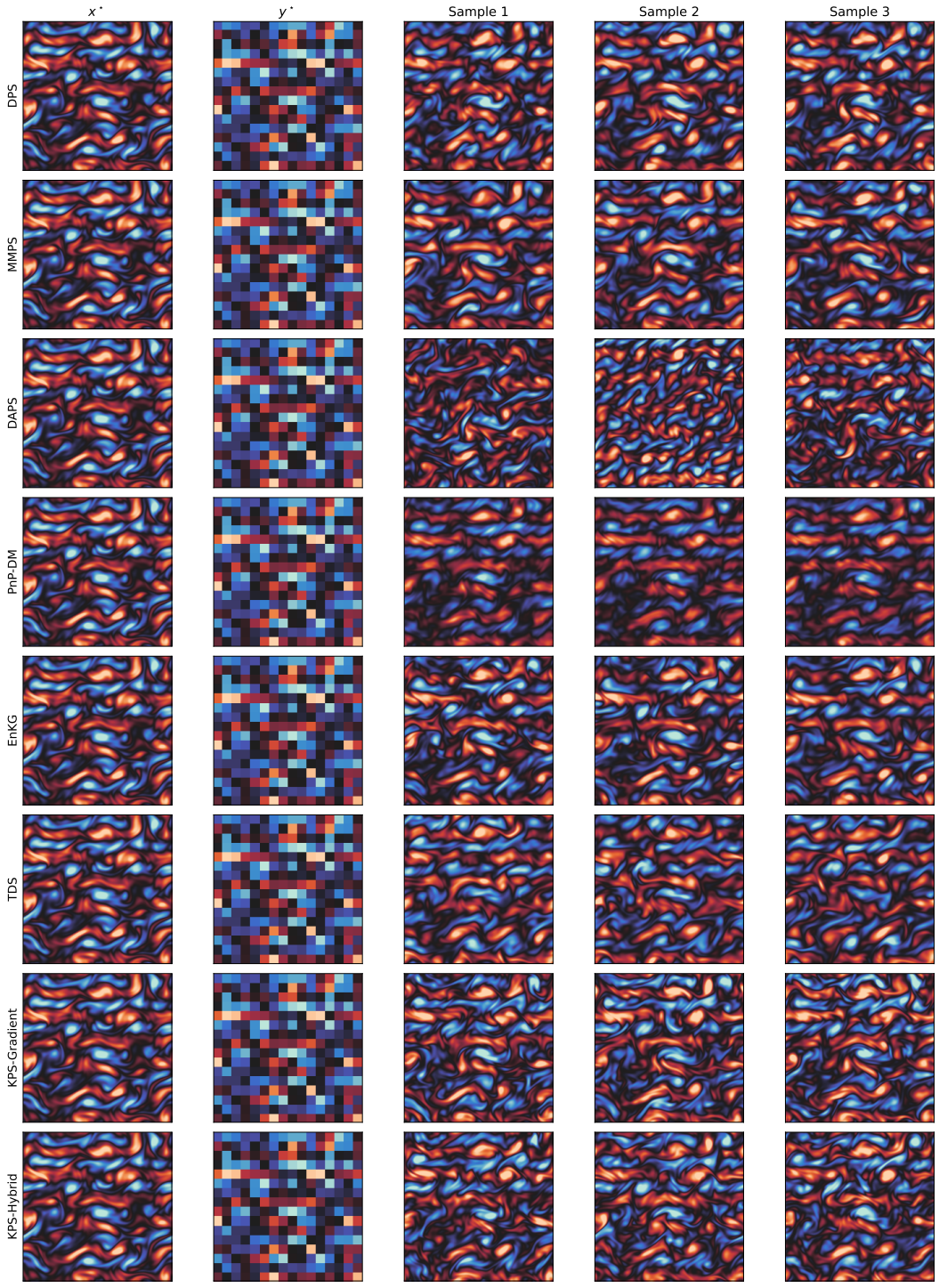

Figure 18: As Figure 17, instance 8.

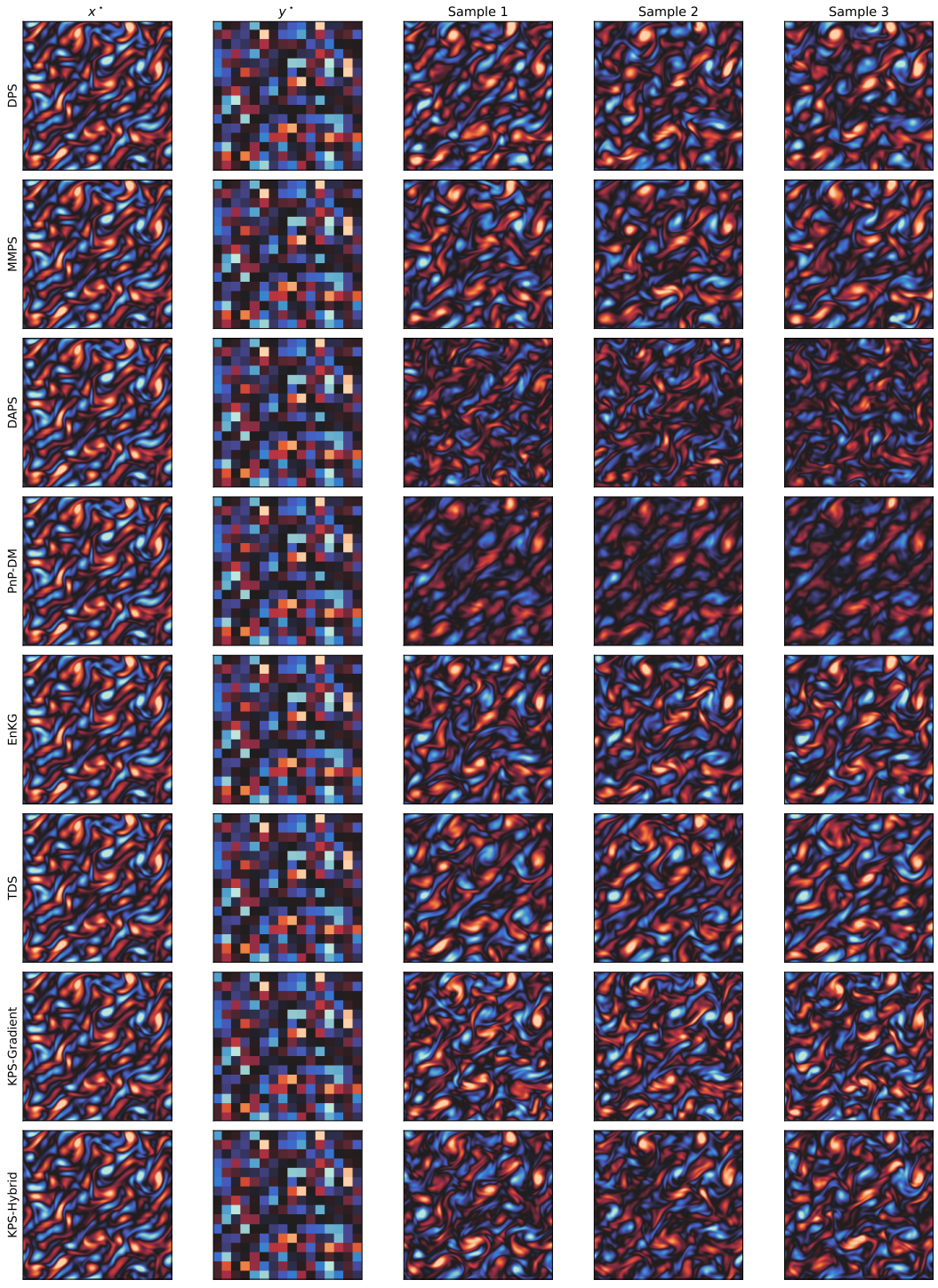

Figure 19: As Figure 17, instance 9.

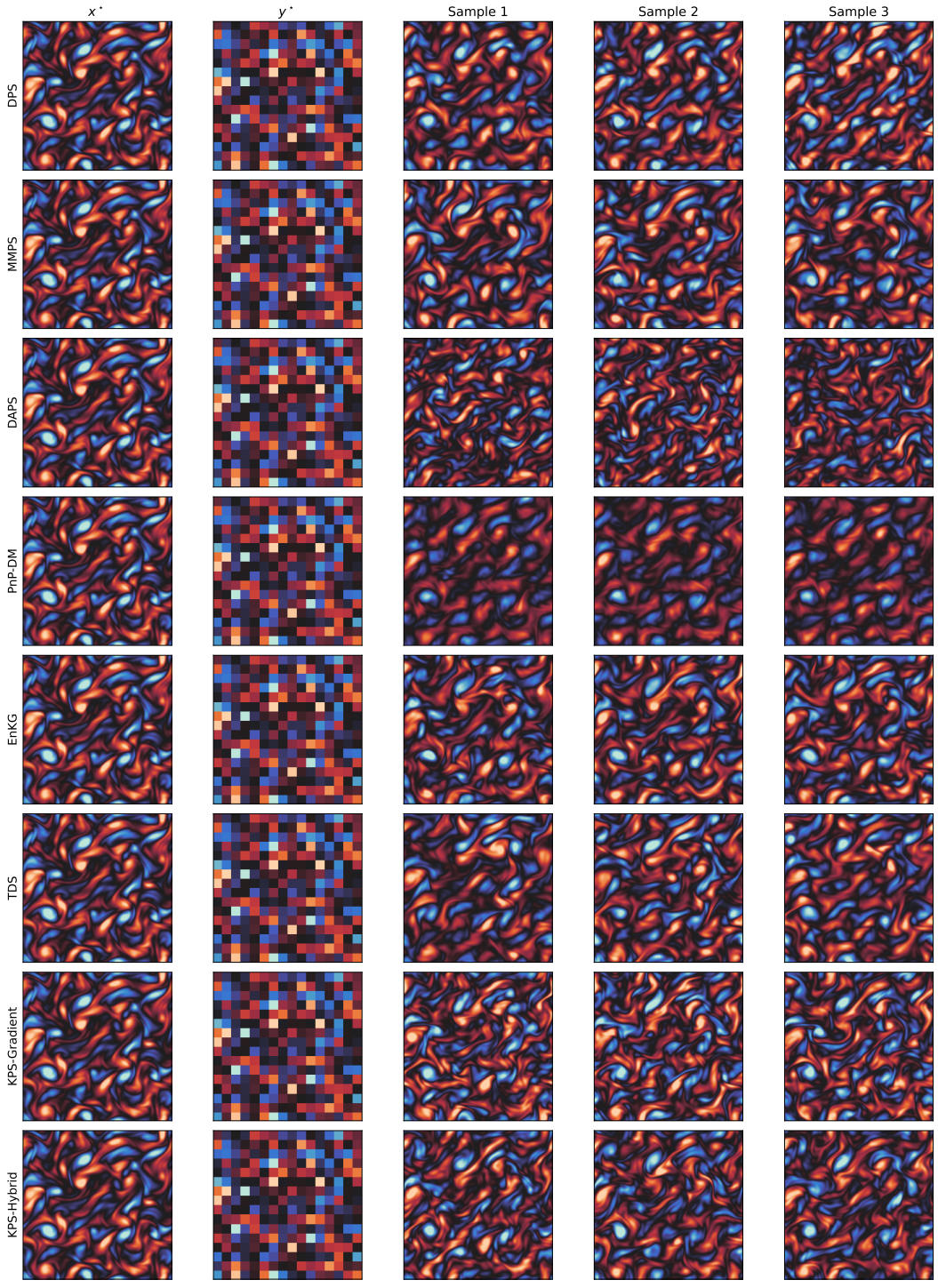

Figure 20: As Figure 17, instance 10.

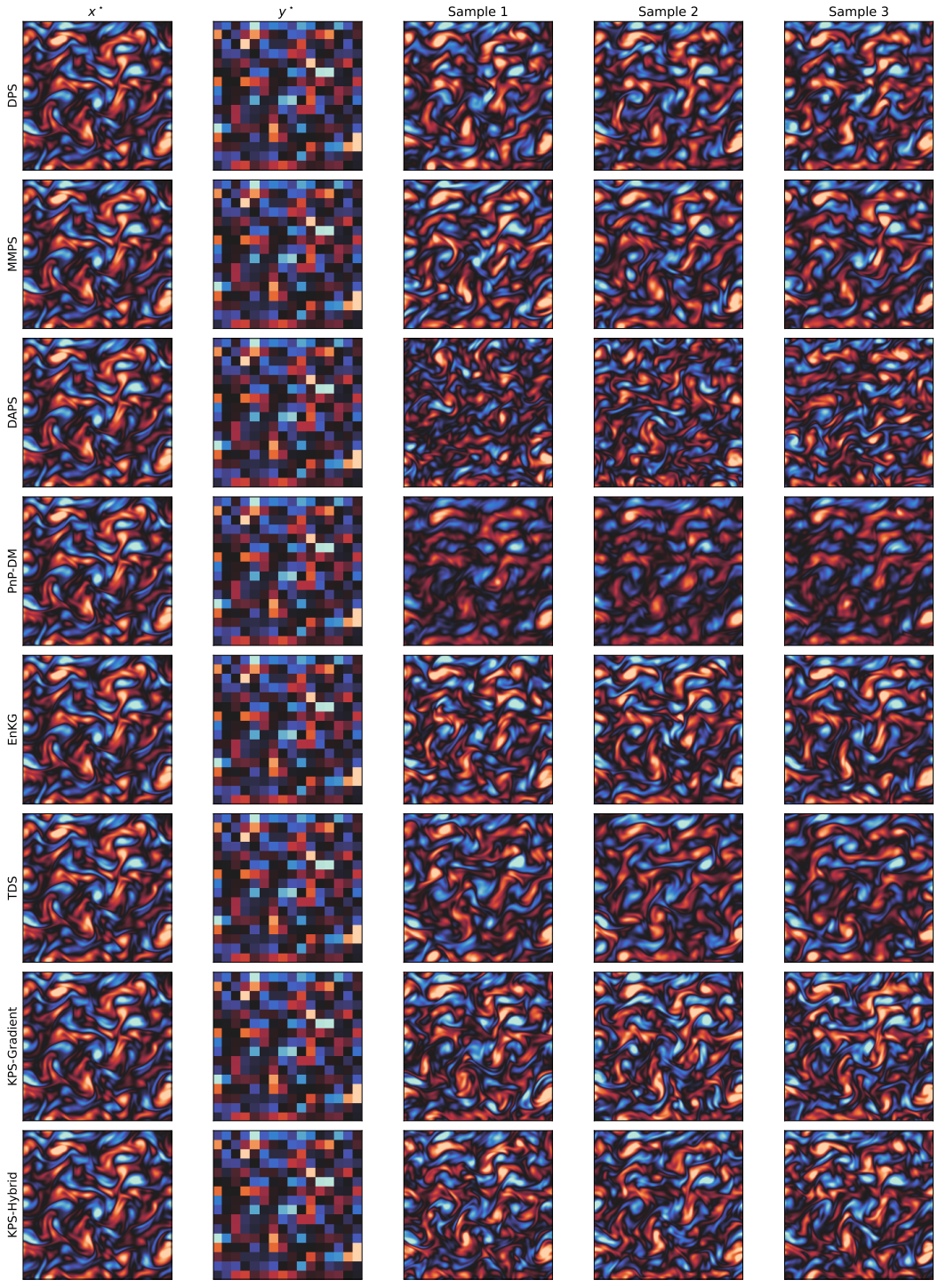

Figure 21: As Figure 17, instance 21.

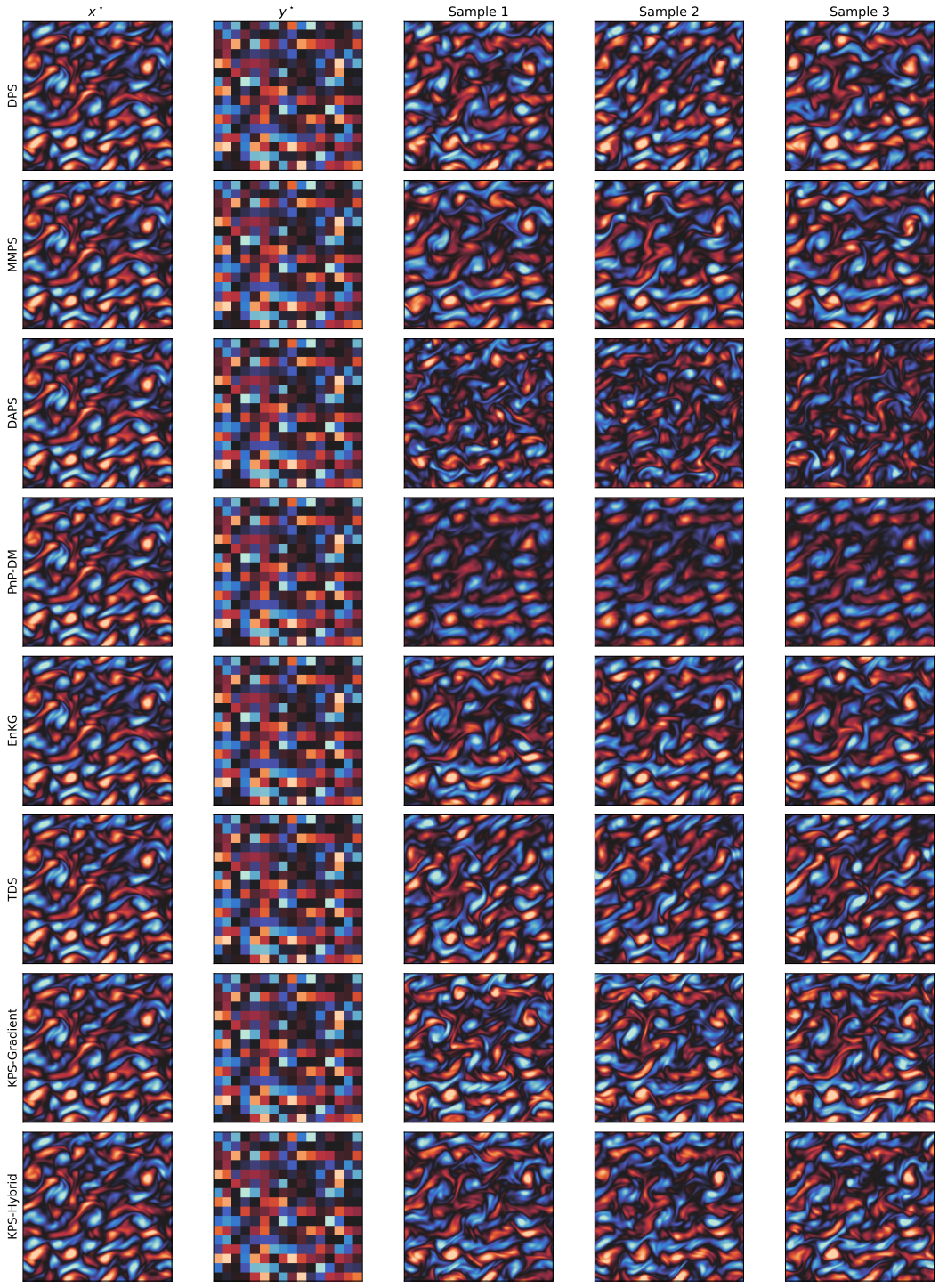

Figure 22: As Figure 17, instance 30.

E Details on baseline methods

In the experiments (Section 5), we compare our method with other training-free posterior sampling algorithms. We describe these methods below following the framework of Section 2, but considering a simplified forward model corresponding to the setting for which these methods were designed:

$$y = \mathcal{A}(x) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \Sigma_y). \tag{44}$$

E.1 Diffusion Posterior Sampling (DPS) (Chung et al., 2023)

Diffusion Posterior Sampling (DPS, Algorithm 3) is one of the most widely used posterior sampling algorithms, notably due to its simplicity. It is based on the approximation of the transition kernel $p(x \mid x_t)$ by a Dirac centered at the conditional expectation $\mathbb{E}[x \mid x_t]$, which yields the following approximation for the likelihood score

$$\nabla_{x_t} \log p(y \mid x_t) \approx \nabla_{x_t} \log p(y \mid \mathbb{E}[x \mid x_t]) \propto -\frac{1}{2} \nabla_{x_t} \|y - \mathcal{A}(\mathbb{E}[x \mid x_t])\|^2_{\Sigma_y}. \tag{45}$$

The conditional expectation can be directly obtained from the denoiser, allowing the approximation of Equation 45 to be computed by automatic differentiation. With a sufficient number of iterations to solve Equation 6, DPS produces samples that are consistent with the observation. However, as observed in the work Xu et al. (2025), DPS tends to concentrate on the modes, resulting in poorly calibrated posterior samples.

Algorithm 3 Diffusion Posterior Sampling (DPS)

Require: observation y, schedule $\{t_k\}_{k=0}^T$, step sizes $\{\zeta_k\}_{k=1}^T$

1: $x_0 \sim \mathcal{N}(0, \sigma_1^2 I)$
2: **for** $k = 0, \dots, T-1$ **do**
3: $\quad \hat{x} \leftarrow d_\theta(t_k, x_k)$
4: $\quad \hat{s}_x \leftarrow \sigma_{t_k}^{-2} (\alpha_{t_k} \hat{x} - x_k)$ ▷ Prior score (Eq. 7)
5: $\quad x'_{k+1} \leftarrow \mathrm{Solver}(t_k, t_{k+1}, x_k, \hat{s}_x)$ ▷ SDE integration (Eq. 6)
6: $\quad x_{k+1} \leftarrow x'_{k+1} - \zeta_{k+1} \nabla_{x_k} \|y - \mathcal{A}(\hat{x})\|^2_{\Sigma_y}$ ▷ DPS guidance (Eq. 45)
7: **end for**
8: **return** x_T

E.2 Moment Matching Posterior Sampling (MMPS) (Rozet et al., 2024)

The Moment Matching Posterior Sampling algorithm (MMPS, Algorithm 4) relies on a more refined approximation of the transition kernel $p(x \mid x_t)$, namely a Gaussian with mean $\mathbb{E}[x \mid x_t]$ and covariance $\mathbb{V}[x \mid x_t]$. Under this approximation, and assuming that the derivative of $\mathbb{V}[x \mid x_t]$ with respect to x_t is negligible, the likelihood score of Equation 9 can be approximated as

$$\nabla_{x_t} \log p(y \mid x_t) \approx \nabla_{x_t}^\top \mathbb{E}[x \mid x_t] A^\top (\Sigma_y + A \mathbb{V}[x \mid x_t] A^\top)^{-1} \big(y - \mathcal{A}(\mathbb{E}[x \mid x_t])\big), \tag{46}$$

where A denotes the Jacobian of $\mathcal{A}$ evaluated at the conditional mean $\mathbb{E}[x \mid x_t]$. Since the covariance matrix $\mathbb{V}[x \mid x_t]$ cannot be explicitly stored in high-dimensional settings, the inverse-vector product of Equation 46 is computed by solving the corresponding linear system using iterative methods such as GMRES (Saad & Schultz, 1986), which only require implicit matrix-vector products. These matrix-vector products are obtained through automatic differentiation, notably using the relation between the first and second moments

$$\mathbb{V}[x \mid x_t] = \alpha_t^{-1} \sigma_t^2 \nabla_{x_t}^\top \mathbb{E}[x \mid x_t]. \tag{47}$$

As with DPS, MMPS can produce samples that are consistent with the observations. However, since it was originally designed for linear forward models, it can become strongly biased and poorly calibrated when the observation operator $\mathcal{A}$ is non-linear.

Algorithm 4 Moment Matching Posterior Sampling (MMPS)

Require: observation y, schedule $\{t_k\}_{k=0}^{T}$

$x_0 \sim \mathcal{N}(0, \sigma_1^2 I)$
for $k = 0, \dots, T-1$ **do**
  $\hat{x} \leftarrow d_\theta(t_k, x_k)$
  $\Sigma_{x|x_k} \leftarrow \alpha_{t_k}^{-1} \sigma_{t_k}^2 \nabla_{x_k}^\top(\hat{x})$ ▷ Prior covariance (Eq. 47)
  $\hat{s}_x \leftarrow \sigma_{t_k}^{-2}(\alpha_{t_k}\hat{x} - x_k)$ ▷ Prior score (Eq. 7)
  $\hat{s}_y \leftarrow \nabla_{x_k}^\top(\hat{x}) A^\top (\Sigma_y + A\Sigma_{x|x_k}A^\top)^{-1}(y - \mathcal{A}(\hat{x}))$ ▷ MMPS likelihood score (Eq. 46)
  $\hat{s}_{x,y} \leftarrow \hat{s}_x + \hat{s}_y$ ▷ Posterior score (Eq. 8)
  $x_{k+1} \leftarrow \mathrm{Solver}(t_k, t_{k+1}, x_k, \hat{s}_{x,y})$ ▷ SDE integration (Eq. 6)
end for
return x_T

E.3 Ensemble Kalman Diffusion Guidance (EnKG) (Zheng et al., 2025a)

When the observation operator $\mathcal{A}$ is non-differentiable, the likelihood score cannot be approximated by automatic differentiation, as done in DPS and MMPS. The Ensemble Kalman Guidance algorithm (EnKG, Algorithm 5) addresses this issue by estimating it from an ensemble of particles $\{x_t^{(i)}\}_{i=1}^N$:

$$\nabla_{x_t^{(i)}} \log p(y \mid x_t^{(i)}) \approx \frac{1}{N} \sum_{j=1}^{N} \langle y^{(j)} - \overline{y}, \Sigma_y^{-1}(y - y^{(i)}) \rangle \left(x_t^{(j)} - \overline{x}_t \right), \tag{48}$$

where $\overline{x}_t$ denotes the ensemble mean of $\{x_t^{(i)}\}_{i=1}^N$, $y^{(i)}$ the observation obtained by integrating Equation 6 up to $t = 0$ from $x_t^{(i)}$ and applying $\mathcal{A}$, and $\overline{y}$ the mean of $\{y^{(i)}\}_{i=1}^N$. This method can be applied on complex scientific problems where the gradient is unavailable, as the Navier-Stokes problem of Zheng et al. (2025b). However, since it is specifically designed to sample MAP solutions, it does not aim to recover the full posterior.

Algorithm 5 Ensemble Kalman Guidance (EnKG)

Require: observation y, schedule $\{t_k\}_{k=0}^{T}$, step sizes $\{\zeta_k\}_{k=1}^{T}$, number of particles N

for $i = 1, \dots, N$ **do**
  $x_0^{(i)} \sim \mathcal{N}(0, \sigma_1^2 I)$
end for
for $k = 0, \dots, T-1$ **do**
  $\overline{x}_k \leftarrow \frac{1}{N} \sum_{i=1}^N x_k^{(i)}$
  for $j = i, \dots, N$ **do**
    $x_{k+1}^{(i)} \leftarrow \mathrm{Solver}\left(t_k, t_{k+1}, x_k^{(i)}\right)$ ▷ SDE integration with prior score (Eq. 6)
    $x^{(i)} \leftarrow \mathrm{Solver}\left(t_k, t_T = 0, x_k^{(i)}\right)$ ▷ SDE integration to $t = 0$ (Eq. 6)
    $y^{(i)} \leftarrow \mathcal{A}(x^{(i)})$
  end for
  $\overline{y} \leftarrow \frac{1}{N} \sum_{i=1}^N y^{(i)}$
  for $i = 1, \dots, N$ **do**
    $g^{(i)} \leftarrow \frac{1}{N} \sum_{j=1}^N \langle y^{(j)} - \overline{y}, \Sigma_y^{-1}(y - y^{(i)}) \rangle (x_k^{(j)} - \overline{x}_k)$ ▷ EnKG likelihood score (Eq. 48)
    $x_{k+1}^{(i)} \leftarrow x_{k+1}^{(i)} + \zeta_{k+1} g^{(i)}$
  end for
end for
return $\{x_T^{(i)}\}_{i=1}^N$

E.4 Gradient Free Meta Flow Maps (MFM-GF) Potaptchik et al. (2026)

Rather than approximating the score of the likelihood through assumptions on the transition kernel $p(x \mid x_t)$, recent flow-map-based methods (Holderrieth et al., 2026; Potaptchik et al., 2026) estimate the conditional expectation in Equation 9 by Monte Carlo (MC) using a map Φ that efficiently generates samples from $p(x \mid x_t)$ in a single function evaluation, avoiding the repeated denoiser evaluations required to solve the generative SDE given in Equation 6:

$$x = \Phi(t, x_t, \varepsilon), \qquad \varepsilon \sim \mathcal{N}(0, I). \tag{49}$$

When the observation operator $\mathcal{A}$ is differentiable, the likelihood score can be estimated from N Monte Carlo samples as (Potaptchik et al., 2026)

$$\nabla_{x_t} \log p(y \mid x_t) \approx \nabla_{x_t} \log \left(\frac{1}{N} \sum_{i=1}^{N} p(y \mid \Phi(t, x_t, \varepsilon^{(i)}) \right), \qquad \varepsilon^{(i)} \sim \mathcal{N}(0, I). \tag{50}$$

Furthermore, when $\mathcal{A}$ is non-differentiable, a gradient-free estimator of posterior score can instead be obtained from the samples (Potaptchik et al., 2026)

$$\nabla_{x_t} \log p(x_t \mid y) \approx \sigma_t^{-2} \left(\alpha_t \frac{\sum_{i=1}^{N} x^{(i)} p(y \mid x^{(i)})}{\sum_{i=1}^{N} p(y \mid x^{(i)})} - x_t \right), \tag{51}$$

where $x^{(i)}$ are the samples generated by the stochastic flow maps. In practice, during the experiment on the latent problem, we found that the number of samples N required for the gradient-free method to perform reliably was prohibitively large. For this reason, the method does not appear in Table 1, although discussed in Section 3.

Algorithm 6 Meta Flow Maps (MFM-GF)

Require: observation y, schedule $\{t_k\}_{k=0}^{T}$, number of MC samples N, stochastic flow maps Φ

$x_0 \sim \mathcal{N}(0, \sigma_1^2 I)$
for $k = 0, \dots, T-1$ **do**
  for $i = 1, \dots, N$ **do**
    $\varepsilon^{(i)} \sim \mathcal{N}(0, I)$
    $x^{(i)} = \Phi(t_k, x_k, \varepsilon^{(i)})$ ▷ Stochastic Flow Maps (Eq. 49)
  end for
  $\hat{s}_{x,y} \leftarrow$ Equation 8 ▷ Posterior score (Eq. 8)
  $x_{k+1} \leftarrow \mathrm{Solver}(t_k, t_{k+1}, x_k, \hat{s}_{x,y})$ ▷ SDE integration (Eq. 6)
end for
return x_T

E.5 Twisted Diffusion Sampler (TDS) (Wu et al., 2023)

Twisted Diffusion Sampler (TDS, Algorithm 7) formulates posterior sampling as a sequential Monte Carlo problem (Doucet & Johansen, 2008). The posterior is represented by a weighted ensemble of particles, obtained through successive sampling, weighting, and resampling steps.

More precisely, the particles $\{x_t^{(i)}\}_{i=1}^{N}$ are first propagated to the next noise level using the likelihood score approximation of DPS (Equation 45), and subsequently weighted according to

$$w^{(i)} \propto \frac{p(x_s^{(i)} \mid x_t^{(i)}) \times p(y \mid \mathbb{E}[x \mid x_s^{(i)}])}{p(x_s^{(i)} \mid x_t^{(i)}, y) \times p(y \mid \mathbb{E}[x \mid x_t^{(i)}])}, \tag{52}$$

where $x_s^{(i)}$ denotes the particle obtained by integrating Equation 6 from t to s with an Euler-Maruyama step initialized at $x_t^{(i)}$, $p(x_s^{(i)} \mid x_t^{(i)})$ the associated probability given by

$$p(x_s^{(i)} \mid x_t^{(i)}) = \mathcal{N}\big(x_s^{(i)} \mid x_t^{(i)} + (s-t)[f_t x_t^{(i)} - g_t^2 \nabla_{x_t^{(i)}} \log p_t(x_t^{(i)})], (t-s) g_t^2 I\big), \tag{53}$$

and $p(x_s^{(i)} \mid x_t^{(i)}, y)$ the associated conditional probability computed using the approximated posterior score instead of the prior score in Equation 53. The particles are then resampled according to these weights, and the procedure is iteratively repeated until $s = 0$.

TDS is asymptotically exact in the limit of infinitely many particles and integration steps. However, like other SMC methods, it suffers from the curse of dimensionality and may require a large number of particles in high-dimensional settings.

Algorithm 7 Twisted Diffusion Sampler (TDS)

Require: observation y, schedule $\{t_k\}_{k=0}^{T}$, number of particles N

for $i = 1, \dots, K$ **do**
  $x_0^{(i)} \sim \mathcal{N}(0, \sigma_1^2 I)$
  $\hat{x}^{(i)} \leftarrow d_\theta(t_0 = 1, x_0^{(i)})$
  $w^{(i)} \propto p(y \mid \hat{x}^{(i)})$
end for
for $k = 0, \dots, T-1$ **do**
  $\{x_k^{(i)}\}_{i=1}^K \sim \text{Multinomial}\left(\{x_k^{(i)}\}_{i=1}^K, \{w^{(i)}\}_{i=1}^K\right)$ ▷ Resampling
  for $i = 1, \dots, K$ **do**
    $\hat{x}^{(i)} \leftarrow d_\theta(t_k, x_k^{(i)})$
    $\hat{s}_x^{(i)} \leftarrow \sigma_{t_k}^{-2}(\alpha_{t_k}\hat{x}^{(i)} - x_k^{(i)})$ ▷ Prior score (Eq. 7)
    $\hat{s}_y^{(i)} \leftarrow -\frac{1}{2}\nabla_{x_k^{(i)}} \|y - \mathcal{A}(\hat{x}^{(i)})\|_{\Sigma_y}^2$ ▷ DPS likelihood score (Eq. 45)
    $\hat{s}_{x,y}^{(i)} \leftarrow \hat{s}_x^{(i)} + \hat{s}_y^{(i)}$ ▷ Posterior score (Eq. 8)
    $x_{k+1}^{(i)} \leftarrow \text{Solver}(t_k, t_{k+1}, x_k^{(i)}, \hat{s}_{x,y}^{(i)})$ ▷ SDE integration (Eq. 6)
    $w^{(i)} \leftarrow \text{Weight}(y, x_k^{(i)}, x_{k+1}^{(i)}, \hat{s}_x^{(i)}, \hat{s}_y^{(i)}, d_\theta)$ ▷ Weighting (Eq. 52)
  end for
end for
return $\{x_T^{(i)}\}_{i=1}^K, \{w^{(i)}\}_{i=1}^K$

E.6 Decoupled Annealing Posterior Sampling (DAPS) (Zhang et al., 2025)

To improve performance on non-linear problems, the Decoupled Annealing Posterior Sampling algorithm (DAPS, Algorithm 8) decouples the different time steps involved when solving Equation 6.

More precisely, given a sample x_t at time $t \in]0, 1]$, DAPS first samples from the transition kernel $p(x \mid x_t)$ by integrating Equation 6 up to $t = 0$. The resulting x is then corrected using Langevin dynamics based on the observation

$$x^{(j+1)} = x^{(j)} + \eta_t \left(\nabla_{x^{(j)}} \log p(x^{(j)} \mid x_t) + \nabla_{x^{(j)}} \log p(y \mid x^{(j)})\right) + \sqrt{2\eta_t}\varepsilon^{(j)}. \tag{54}$$

The likelihood score of Equation 54 is computed analytically from Equation 44, while the score of prior relies on a Gaussian assumption

$$p(x \mid x_t) \approx \mathcal{N}(\mathbb{E}[x \mid x_t], r_t^2 I), \tag{55}$$

where the variance r_t^2 is a hyperparameter specified using heuristics. After N iterations of Equation 54, the resulting sample $x^{(N)}$ can be considered as a sample from the conditional transition kernel $p(x \mid x_t, y)$ and be propagated to any time $s < t$ using the transition kernel

$$x_s \sim \mathcal{N}(\alpha_s x^{(N)}, \sigma_s^2 I) = p(x_s \mid x^{(N)}). \tag{56}$$

Algorithm 8 Decoupled Annealing Posterior Sampling (DAPS)

Require: observation y, schedule $\{t_k\}_{k=0}^{T}$, number of Langevin steps N, step sizes $\{\eta_k\}_{k=0}^{T-1}$

$x_0 \sim \mathcal{N}(0, \sigma_1^2 I)$
for $k = 0, \dots, T-1$ **do**
  $\hat{x}^{(0)} \leftarrow \text{Solver}(t_k, t_T = 0, x_k)$ ▷ SDE integration to $t = 0$ (Eq. 6)
  for $j = 0, \dots, N-1$ **do**
    $\hat{s}_x^{(j)} \leftarrow \nabla_{\hat{x}^{(j)}} \log p(\hat{x}^{(j)} \mid x_k)$ ▷ From DAPS prior approximation (Eq. 55)
    $\hat{s}_y^{(j)} \leftarrow \nabla_{\hat{x}^{(j)}} \log p(y \mid \hat{x}^{(j)})$
    $\varepsilon^{(j)} \sim \mathcal{N}(0, I)$
    $\hat{x}^{(j+1)} \leftarrow \hat{x}^{(j+1)} + \eta_k \left(\hat{s}_x^{(j)} + \hat{s}_y^{(j)}\right) + \sqrt{2\eta_k}\varepsilon^{(j)}$ ▷ Langevin dynamics (Eq. 54)
  end for
  $x_{k+1} \sim \mathcal{N}\left(\alpha_{t_{k+1}} \hat{x}^{(N)}, \sigma_{t_{k+1}}^2 I\right)$ ▷ Next noise level (Eq. 56)
end for
return x_T

E.7 Plug-and-Play Diffusion Models (PnP-DM) (Wu et al., 2024)

Following a similar principle as DAPS, Plug-and-Play Diffusion Models (PnP-DM, Algorithm 9) uses Langevin dynamics to update an auxiliary variable coupled to the variable of interest.

More precisely, an auxiliary variable z is initialized from $\mathcal{N}(0, I)$ and updated using N iterations of a Langevin dynamics to incorporate the observation y while remaining close to its initial state $z^{(0)}$. The resulting $z^{(N)}$ is then scaled by a coefficient α_s associated with a time $s \in]0, 1]$, yielding an x_s that serves as the initial state for solving Equation 6 from $t = s$ to $t = 0$. The resulting x can then be used to initialize a new Langevin process, and the procedure is repeated for successive time steps.

Algorithm 9 Plug-and-Play Diffusion Models (PnP-DM)

Require: observation y, initialization x, schedule $\{t_k\}_{k=0}^{T}$, Langevin steps N, step sizes $\{\eta_k\}_{k=0}^{T-1}$

$x_0 \leftarrow x$
for $k = 0, \dots, T-1$ **do**
  $z_k^{(0)} \leftarrow x_k$
  $\rho_k \leftarrow \frac{\sigma_{t_k}}{\alpha_{t_k}}$
  for $j = 0, \dots, N-1$ **do** ▷ Likelihood step
    $\hat{s}_y^{(j)} \leftarrow -\frac{1}{2}\nabla_{z_k^{(j)}} \|y - \mathcal{A}(z_k^{(j)})\|_{\Sigma_y}^2$
    $\varepsilon^{(j)} \sim \mathcal{N}(0, I)$
    $z_k^{(j+1)} \leftarrow z_k^{(j+1)} + \eta_k \hat{s}_y^{(j)} - \frac{\eta_k}{\rho_k^2}(z_k^{(j)} - x_k) + \sqrt{2\eta_k}\varepsilon^{(j)}$ ▷ Langevin dynamics
  end for
  $x_k \leftarrow \alpha_{t_k} z_k^{(N)}$
  $x_{k+1} \leftarrow \text{Solver}(t_k, t_T = 0, x_k)$ ▷ Prior step (Eq. 6)
end for
return x_T