Flow Map Denoisers: Traversing the Distortion-Perception Plane for Inverse Problems

Nicolas Zilberstein
Rice University
nzilberstein@rice.edu

Morteza Mardani
NVIDIA Inc.
mmardani@nvidia.com

Santiago Segarra
Rice University
segarra@rice.edu

Abstract

Image restoration faces a fundamental tradeoff: methods that minimize error produce blurry reconstructions, while those that maximize perceptual quality yield sharp but less faithful images. Existing approaches either commit to a single operating point on this distortion–perception (DP) frontier or require paired-data supervision, auxiliary models, or hyperparameter tuning of the sampler to access different points. We show that flow map models, a recent extension of flow matching for few-step sampling that learns an average field, implicitly define a one-parameter family of denoisers that continuously traverses the DP plane. The lookahead parameter t acts as a control knob between the MMSE and perceptual regimes. For isotropic Gaussian targets, we prove that varying t exactly recovers the optimal DP frontier; for natural images, we observe similar behavior empirically. Within a Plug-and-Play solver, the same mechanism extends to general inverse problems, where the lookahead acts as a regularization parameter that controls a tradeoff between perceptual alignment and data consistency. Despite the lack of exact optimality guarantees in this setting, a single trained flow map traverses the DP tradeoff, being competitive with specialized baselines at both extremes and with training-free DP-traversal methods. Extensive experiments on CelebA (128×128) and AFHQ (256×256) across several linear and nonlinear inverse tasks validate our findings. Code is available at `https://github.com/nzilberstein/Flow-map-denoisers`.

1 Introduction

Image restoration is a highly ill-posed inverse problem: a single degraded observation can correspond to a vast set of plausible clean images. Standard regression approaches that minimize mean squared error (MSE) approximate the conditional expectation (Ongie et al., 2020), achieving minimum distortion at the cost of over-smoothed reconstructions that lack fine detail. In contrast, posterior sampling techniques (Kawar et al., 2021) produce realistic-looking solutions at the cost of higher distortion. This inherent compromise is formalized by the *distortion-perception (DP) tradeoff* (Blau and Michaeli, 2018; Freirich et al., 2021). Because the ideal balance between low distortion and high perceptual quality is strictly user-dependent, designing a single model capable of traversing the DP plane at inference time remains a central challenge in computational imaging.

Most existing methods commit to a *single operating point in the DP plane*, either targeting minimum distortion through supervised regression (Dong et al., 2015; Terris et al., 2026), or minimum perception by leveraging recent generative models as priors (Pokle et al., 2024; Chung et al., 2023; Kadkhodaie and Simoncelli, 2021) to draw samples from the posterior. A few recent works based

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

on data-dependent flow models aim to traverse the DP plane (Ohayon et al., 2025; Delbracio and Milanfar, 2023; Albergo et al., 2024; Zhu et al., 2026), by explicitly interpolating between measurement-dependent estimators (e.g., the conditional mean) and the clean image. Other training-free methods move along the DP plane by tuning the sampler, e.g., scaling the injected noise (Zhu et al., 2023; Wang et al., 2025), interpolating between guidance terms (Garber and Tirer, 2024), or re-noising a MAP estimate before posterior sampling (Zhang et al., 2026). Unlike these approaches, which require external mechanisms, such as interpolation, solver hyperparameters, or retraining, to move along the DP frontier, we show that this tradeoff is already implicitly encoded in the dynamics of flow maps.

In this work, we propose a unified approach based on *flow maps* (Boffi et al., 2025b,a; Geng et al., 2025; Sabour et al., 2025b), which unifies all operating points along the DP plane within a single model. While prior work primarily uses flow maps to accelerate sampling (Gülle et al., 2026; Sabour et al., 2025a; Spagnoletti et al., 2025), we instead leverage them to *traverse the DP plane*. Our key observation is that the lookahead t selects a point along a continuum of estimators—termed *average denoisers*—implicitly encoded within a single trained flow map, thus acting as a *knob* to control the characteristics of the restored image. This perspective reveals DP control as an intrinsic property of the learned flow map, eliminating the need for multiple models, auxiliary networks, or heuristic solver choices.

For isotropic Gaussian targets, we prove that this family spans the entire DP frontier, recovering the MMSE estimator at one extreme and a perfect-perception estimator at the other, tracing exactly the optimal DP curve (Freirich et al., 2021). While this exact theoretical correspondence strictly holds for Gaussian denoising problems, we empirically observe similar behavior for natural images. When extending this approach to general, non-Gaussian inverse problems, traversing the lookahead serves as an approximation to the true DP frontier; here, the lookahead acts as a regularization parameter that controls a tradeoff combining perceptual alignment with data-consistency effects, rather than exactly reproducing the denoising DP frontier. By embedding these denoisers within a Plug-and-Play (PnP) framework (Venkatakrishnan et al., 2013; Romano et al., 2017), we leverage this property to obtain a versatile solver that effectively traverses the DP plane across a wide range of inverse problems. Fig. 1 illustrates this behavior on a 2D Gaussian mixture model.

Overall, our contributions are:

- **Flow maps as a continuum of DP estimators.** We reinterpret flow maps as a family of denoisers indexed by a lookahead parameter t, enabling continuous traversal of the DP plane with a single model.
- **Exact optimality in the Gaussian case.** We show that this family recovers the *optimal* DP curve for isotropic Gaussian targets, establishing a direct link between flow maps and DP theory.
- **PnP reconstruction with continuous control.** We embed these denoisers into a plug-and-play framework, yielding a unified solver that traverses the DP plane without retraining, and enables high-perceptual reconstructions at a fraction of the cost of posterior sampling methods.
- **Empirical validation.** On CelebA and AFHQ inverse problems (inpainting, motion deblurring, super-resolution, Gaussian deblurring), a single trained flow map is competitive with specialized baselines at both DP endpoints and traces a smooth curve in between, without retraining, auxiliary networks, or retuning the solver across operating points.

Notation. Bold symbols (e.g., $\mathbf{x}$) denote vectors, and bold uppercase symbols (e.g., $\mathbf{H}$) denote matrices. Mappings such as the flow map $X_{s,t}$ and denoisers $D_{s,t}$ are written in standard (non-bold) font.

2 Background

2.1 Flow matching and flow maps

Stochastic interpolants. Stochastic interpolants (Albergo et al., 2025; Lipman et al., 2023) define a family of stochastic processes that interpolate between a source distribution p_0 (typically $\mathcal{N}(0, \mathbf{I})$) and a target distribution p_1 (the data distribution):

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_1, \tag{1}$$

where $\alpha, \beta : [0, 1] \to \mathbb{R}$ are continuously differentiable functions satisfying the boundary conditions $\alpha_0 = 1, \alpha_1 = 0$ and $\beta_0 = 0, \beta_1 = 1$. A common choice is the linear interpolant $\alpha_t = 1 - t$ and

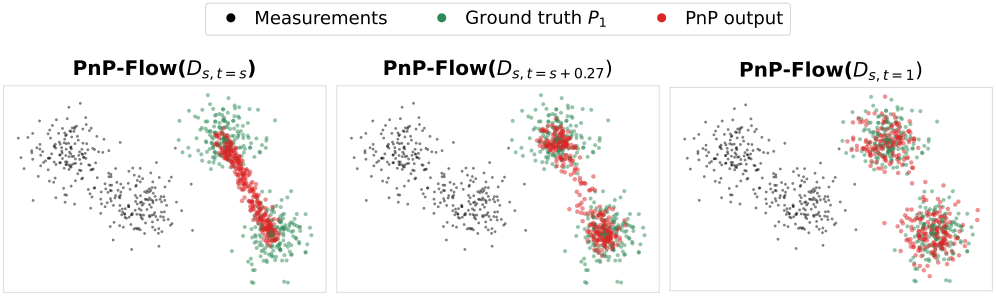


Figure 1: PnP with average denoisers on a 2D mixture-of-Gaussians inverse problem ($\mathbf{y} = \mathbf{H}\mathbf{x} + \sigma\boldsymbol{\epsilon}$, with $\mathbf{H}$ a rotation+scaling operator and $\sigma = 0.3$). $D_{s,t}$ denotes the average denoiser (defined formally in Section 3.1). The lookahead t controls the DP trade-off: at $t = s$ the denoiser is MMSE and PnP reconstructions concentrate at the posterior mean, which for this symmetric bimodal posterior lies between the two modes, yielding low distortion but poor perceptual quality; as $t \to 1$, the lookahead progressively restores the bimodal structure, recovering samples from both modes at the cost of higher distortion; details of this experiment are in Appendix B.4.

$\beta_t = t$, yielding $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$. The marginal velocity field $\mathbf{v}(\mathbf{x}, t) = \mathbb{E}[\dot{\mathbf{x}}_t \mid \mathbf{x}_t = \mathbf{x}]$ yields an ODE $\dot{\mathbf{x}}_t = \mathbf{v}_t(\mathbf{x}_t)$, defining a continuous-time transport map from the source to the target distribution and which can be used to draw samples from p_1. In practice, a neural network $\mathbf{v}_\theta(\mathbf{x}, t)$ is trained to approximate v_t via the mean square error loss:

$$\mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1}\left[\left\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\dot{\alpha}_t\mathbf{x}_0 + \dot{\beta}_t\mathbf{x}_1)\right\|_2^2\right], \tag{2}$$

which plays a role analogous to denoising score matching (Vincent, 2011) in diffusion models (Song et al., 2021). For the linear interpolant, the velocity field relates directly to the Minimum Mean Squared Error (MMSE) denoiser – *instantaneous denoiser* – via Tweedie's formula (Robbins, 1956; Miyasawa et al., 1961):

$$D_t(\mathbf{x}) := \mathbb{E}[\mathbf{x}_1 \mid \mathbf{x}_t = \mathbf{x}] = \mathbf{x} + (1-t)\mathbf{v}(\mathbf{x}, t). \tag{3}$$

Flow maps. Sampling from p_1 using the learned velocity field requires numerical integration of the ODE, typically with many function evaluations. Flow maps (Boffi et al., 2025b,a; Geng et al., 2025) bypass this integration by training a model to directly predict the average over a finite interval $[s, t]$:

$$X_{s,t}(\mathbf{x}) = \mathbf{x} + (t-s)\,\mathbf{v}(\mathbf{x}, s, t), \tag{4}$$

where $\mathbf{v}(\mathbf{x}, s, t)$ is the *average velocity* over the interval $[s, t]$, which reduces to the instantaneous velocity on the diagonal, $\mathbf{v}(\mathbf{x}, t, t) = \mathbf{v}(\mathbf{x}, t)$. A single network $\mathbf{v}_\theta(\mathbf{x}, s, t)$ is trained to predict this average velocity, conditioned on both the current time s and the target time t. Among the self-distillation objectives for flow maps (Boffi et al., 2025b), we use the Lagrangian self-distillation (LSD) loss, which combines the diagonal loss (2) applied to the slice $\mathbf{v}_\theta(\cdot, t, t)$ with the off-diagonal term

$$\mathcal{L}_{\mathrm{LSD}}(\theta) = \int_0^1 \int_0^t \mathbb{E}_{\mathbf{x}_0,\mathbf{x}_1}\left[\left\|\partial_t \hat{X}_{s,t}(\mathbf{x}_s) - \mathbf{v}_\theta\left(\hat{X}_{s,t}(\mathbf{x}_s), t, t\right)\right\|^2\right] \mathrm{d}s\,\mathrm{d}t, \tag{5}$$

where $\hat{X}_{s,t}(\mathbf{x}) = \mathbf{x} + (t-s)\,\mathbf{v}_\theta(\mathbf{x}, s, t)$. The velocity on the right-hand side is the diagonal slice of the *same* network, so no pre-trained teacher is required. We use LSD because it gave the most stable training and the best visual quality among the objectives we tried; further background is in Appendix A.1.

2.2 Inverse problems and Plug-and-Play methods

We consider inverse problems of the form

$$\mathbf{y} = f(\mathbf{x}) + \mathbf{n}, \tag{6}$$

where $f(\cdot)$ is a known (possibly non-linear) degradation operator and $\mathbf{n}$ is additive noise, typically Gaussian, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Two main families of methods combine flow or diffusion priors with data-consistency updates (Daras et al., 2024; Chung et al., 2025; Zheng et al., 2025): guidance-based methods (Zhang et al., 2024; Pokle et al., 2024; Chung et al., 2023; Song et al., 2022) and plug-and-play (optimization-based) methods (Ben-Hamu et al., 2024; Martin et al., 2025). We focus on the latter and defer a discussion of guidance-based approaches to Appendix A.2.

Plug-and-Play methods. Plug-and-Play (PnP) algorithms (Venkatakrishnan et al., 2013; Romano et al., 2017) originate from proximal optimization methods for solving inverse problems of the form $\min_{\mathbf{x}} g(\mathbf{x}) + h(\mathbf{x})$. In particular, forward–backward splitting (FBS) alternates between a gradient step on the data-fidelity term, $\mathbf{z}^k = \mathbf{x}^k - \lambda \nabla g(\mathbf{x}^k)$, and a proximal step on the regularizer, $\mathbf{x}^{k+1} = \mathrm{prox}_{\lambda h}(\mathbf{z}^k)$, where the proximal operator is defined as $\mathrm{prox}_{\lambda h}(\mathbf{z}) := \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|^2 + \lambda h(\mathbf{x})$. Computing $\mathrm{prox}_{\lambda h}$ corresponds to solving a denoising problem.

Flow and diffusion models as PnP priors. Following the PnP principle (Venkatakrishnan et al., 2013; Romano et al., 2017), recent works instantiate this proximal step with a learned denoiser D_t, yielding the iteration $\mathbf{x}^{k+1} = D_t(\mathbf{z}^k)$ (Zhu et al., 2023; Martin et al., 2025; Mardani et al., 2024; Zilberstein et al., 2025; Renaud et al., 2024; Laumont et al., 2022, 2023; Hu et al., 2024). This approach leverages implicit priors learned by generative models while retaining the flexibility of PnP for arbitrary forward operators. Because PnP is derived from proximal splitting, its output is biased toward a MAP-type solution, which favors low distortion at the expense of perceptual quality and yields over-smoothed reconstructions. In Section 3, we show that replacing the instantaneous denoiser with the average denoiser of a flow map enables continuous control over this tradeoff.

2.3 Distortion-Perception Tradeoff

The standard evaluation of image restoration methods relies on average distortion $D = \mathbb{E}[\Delta(\mathbf{x}, \hat{\mathbf{x}})]$ (e.g., MSE or LPIPS), which measures the discrepancy between the ground truth $\mathbf{x}$ and its estimate $\hat{\mathbf{x}}$. However, since reconstructions should also appear natural to humans, methods are additionally assessed by perceptual quality. In practice, this is approximated by a *perceptual index* $P = d(p_{\mathbf{x}}, p_{\hat{\mathbf{x}}})$, where $p_{\hat{\mathbf{x}}}$ denotes the marginal distribution of the reconstructions and d is a divergence (e.g., the Wasserstein distance) between the true and reconstructed distributions; it is estimated via surrogates such as FID (Heusel et al., 2017) or KID (Bińkowski et al., 2018).

Blau and Michaeli (2018) showed that distortion and perception are fundamentally at odds, defining the *distortion–perception (DP) tradeoff*. Formally, the DP function is the smallest distortion achievable by any estimator at a given perception level,

$$D(P) = \min_{p_{\hat{\mathbf{x}}|\mathbf{y}}} \mathbb{E}\big[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\big] \quad \text{s.t.} \quad W_2(p_{\mathbf{x}}, p_{\hat{\mathbf{x}}}) \le P,$$

where we use the MSE distortion and the Wasserstein-2 distance for perception; we refer to the graph of $D(P)$ as the *optimal DP curve*. For this choice, Freirich et al. (2021) showed that $D(P) = D^* + \big(\max\{P^* - P, 0\}\big)^2$, where $D^* = \mathbb{E}[\|\mathbf{x} - \mathbf{x}^*\|^2]$ is the MMSE distortion and $P^* = W_2(p_{\mathbf{x}}, p_{\mathbf{x}^*})$ is the perceptual gap of the posterior mean $\mathbf{x}^* = \mathbb{E}[\mathbf{x}|\mathbf{y}]$. Note that P constrains the *marginal* distribution of the reconstructions, which is also what FID and KID measure. The optimal estimator at each perception level is given by the interpolation

$$\hat{\mathbf{x}}_P = \left(1 - \frac{P}{P^*}\right) \hat{\mathbf{x}}_0 + \frac{P}{P^*} \mathbf{x}^*, \tag{7}$$

where $\hat{\mathbf{x}}_0$ is a perfectly perceptual estimator. This result implies that the entire DP frontier can be recovered by interpolating between two endpoints: the MMSE estimator $\mathbf{x}^*$ and a perceptually optimal estimator $\hat{\mathbf{x}}_0$. Existing approaches access intermediate operating points by interpolating between separate estimators, by task-specific training (Delbracio and Milanfar, 2023; Ohayon et al., 2025), or by tuning the sampler (Wang et al., 2025; Zhang et al., 2026). In contrast, we seek a *single* model whose conditioning directly parameterizes this continuum.

3 Average denoisers: traversing the DP plane

In this section we define and analyze the *average denoiser* associated to flow maps and its connection to the DP tradeoff (Section 3.1).

3.1 Flow maps as denoisers

Recall we seek a family of estimators parametrized by a *single model* that interpolates between $P = P^*$ (lowest distortion) and $P = 0$ (highest perceptual quality). A natural candidate is the

denoiser implicitly defined by the flow map (Section 2.1). The same object was independently introduced by Lee et al. (2026); Lu et al. (2026) for one-step language modeling on discrete data; here we study it in the continuous setting and show it parametrizes the DP frontier.

Definition 1 (Average denoiser) *Given a flow map with average velocity* $\mathbf{v}(., s, t)$*, the* average denoiser *(or two-time denoiser) is:*

$$D_{s,t}(\mathbf{x}) := \mathbf{x} + (1 - s)\mathbf{v}(\mathbf{x}, s, t). \tag{8}$$

This definition has the same form as the instantaneous denoiser (3), but replaces the instantaneous velocity $\mathbf{v}(., s)$ with the average velocity $\mathbf{v}(., s, t)$ over the interval $[s, t]$. This modification has an important consequence: the parameter t now controls where $D_{s,t}(\mathbf{x})$ sits on the DP plane. At $t = s$, $D_{s,s}$ is the MMSE denoiser (lowest distortion); at $t = 1$, $D_{s,1}(\mathbf{x}) = X_{s,1}(\mathbf{x})$ transports the noisy input to the end of the flow, so its outputs follow the data distribution (highest perceptual quality); intermediate t interpolates between the two. Importantly, both s and t are inputs of $\mathbf{v}_\theta(\mathbf{x}, s, t)$, so a *single trained model* produces a whole family of denoisers with different characteristics, as we show in the following section.

3.2 Traversing the DP plane with flow maps

3.2.1 Warm-up: Gaussian case (1D analysis)

We start with the Gaussian case, which renders a closed-form expression of the average denoiser in (8). We derive the results in the scalar case ($x \in \mathbb{R}$). They extend to isotropic targets $\mathcal{N}(\mathbf{0}, \sigma_p^2\mathbf{I})$ by applying the analysis component-wise. Let $p_1 = \mathcal{N}(0, \sigma_p^2)$, with $\sigma_p^2 \leq 1$ (a normalization convention, not a requirement; see Appendix D.2). Following the stochastic interpolant formulation in Section 2.1, the marginal variance at noise level t is $\sigma_t^2 = t^2\sigma_p^2 + (1-t)^2$, and the instantaneous denoiser is *linear*, given by $\mathbb{E}[x_1 \mid x_t = x] = a_t x$, with the scalar gain $a_t = t\sigma_p^2/\sigma_t^2$. Using Tweedie's relationship in (3), the instantaneous velocity field is also linear, $v(x, t) = b_t x$, with $b_t = [t(\sigma_p^2 + 1) - 1]/\sigma_t^2$.

Flow map and average denoiser. The linearity of the instantaneous velocity yields a scalar flow map $x_t = \Phi(t, s)\, x_s$ with $\Phi(t, s) = \sigma_t/\sigma_s$, obtained by integrating $\int_s^t b_\tau\, d\tau = \log(\sigma_t/\sigma_s)$ (see Appendix D.2). Following Definition 1, the average denoiser reduces to $D_{s,t}(x_s) = \Lambda(s, t)\, x_s$ where:

$$\Lambda(s, t) = \frac{(1 - s)\sigma_t - (1 - t)\sigma_s}{\sigma_s(t - s)}. \tag{9}$$

This gain satisfies $\Lambda(s, s) = a_s$, the MMSE gain (lowest distortion), and $\Lambda(s, 1) = \sigma_p/\sigma_s$, the gain that maps $x_s \sim \mathcal{N}(0, \sigma_s^2)$ exactly onto p_1 (highest perceptual quality), and is strictly increasing in t (see Appendix D.2).

Optimal DP estimators in the Gaussian case. For a Gaussian target, the optimal DP estimator in (7) is also linear, given by $\hat{x}_P = \Gamma_s(\alpha)\, x_s$, with the gain given by

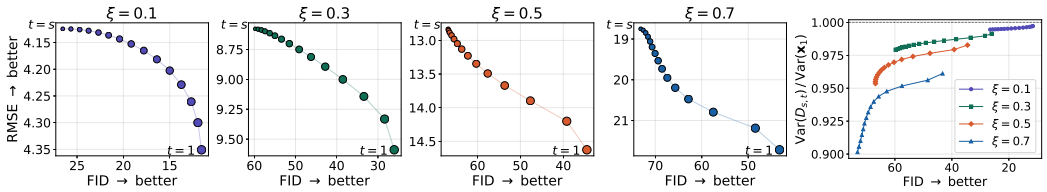
$$\Gamma_s(\alpha) = \alpha\, a_s + (1 - \alpha)\, \sigma_p/\sigma_s. \tag{10}$$

The parameter $\alpha = P/P^* \in [0, 1]$ parametrizes the perception level, interpolating linearly between the MMSE gain a_s and the perfect-perception gain σ_p/σ_s; both gains, and hence Γ_s, depend on the noise level s. Given this particular instance of the DP estimator, we can establish a direct relationship between the flow map and the DP estimator in the Gaussian case:

Theorem 1 (Exact optimality) *Let* $p_1 = \mathcal{N}(0, \sigma_p^2)$ *with* $\sigma_p^2 \leq 1$*, and let* $\Lambda(s, t)$ *be the average denoiser gain* (9) *associated with the true flow map. Then for every* $s \in (0, 1)$ *and* $t \in [s, 1]$*, there exists a perception level*

$$\alpha(s, t) = \frac{\sigma_p/\sigma_s - \Lambda(s, t)}{\sigma_p/\sigma_s - a_s} \in [0, 1] \tag{11}$$

such that $\Lambda(s, t) = \Gamma_s(\alpha(s, t))$*. Moreover,* $\alpha(s, t)$ *decreases monotonically from* 1 *at* $t = s$ *(MMSE,* $P = P^*$*) to* 0 *at* $t = 1$ *(perfect perception,* $P = 0$*). Consequently, the average denoiser* $D_{s,t}$ *traces the optimal DP curve exactly as* t *varies in* $[s, 1]$*.*



(a) DP tradeoff curves (b) Var. vs FID

Figure 2: Distortion–perception tradeoff and variance restoration for image denoising on CelebA 128×128. **(a)** RMSE vs. FID at different noise levels ξ; each curve is swept by varying the lookahead t of a single trained model. **(b)** Variance ratio vs. FID; the near-collapse of the curves across noise levels suggests a common relationship between variance restoration and perceptual quality.

The proof is deferred to Appendix D.2. In summary, both $\Lambda(s,t)$ and the optimal DP gain $\Gamma_s(\alpha)$ are continuous monotonic functions that share the same endpoints: a_s (MMSE, lowest distortion) and σ_p/σ_s (highest perceptual quality). Hence, inverting the affine relation $\Gamma_s(\alpha) = \Lambda(s,t)$ yields a bijection $\alpha(s,t)$ between the lookahead and the perception level, so the average denoiser exactly realizes the optimal DP estimator at every intermediate operating point.

For anisotropic targets $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, both endpoints remain exactly optimal, but intermediate lookaheads are exactly optimal only for a flat spectrum (Appendix D.2.3). We now turn to more general distributions.

3.3 Beyond the Gaussian case: image denoising experiments

For general image distributions, the average denoiser no longer admits a closed-form characterization. While exact DP optimality is not expected beyond the Gaussian case, the structural insight underlying Theorem 1 should still hold: as t varies from s to 1, the output of $D_{s,t}$ is expected to interpolate smoothly between the MMSE estimator (low distortion, low perception quality) and a perceptually-aligned estimator (higher distortion, improved perception). Consequently, we expect (i) smooth, monotonic DP curves parameterized by t, and (ii) convex trajectories consistent with the optimal DP function (Section 2.3), which we verify empirically below. While this section focuses on image data, we further support this intuition for a 2D mixture of Gaussians (MoG) in Appendix D.3, where the DP curve can be computed explicitly.

Setup. We train a flow map network $\mathbf{v}_\theta(\mathbf{x}, s, t)$ on CelebA 128×128 (Appendix B) and evaluate $D_{s,t}(\mathbf{x}_s)$ at multiple lookahead values $t \in [s, 1]$, where $t = s$ recovers the MMSE denoiser D_s and $t = 1$ reaches the endpoint of the flow. Since the stochastic interpolant (1) at time s takes the form $\mathbf{x}_s = s\,\mathbf{x}_1 + (1-s)\,\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, applying the average denoiser $D_{s,t}$ to $\mathbf{x}_s$ corresponds to Gaussian denoising: recovering the clean signal $\mathbf{x}_1$ from an observation where the signal is scaled by s and corrupted by noise of standard deviation $1 - s$. We parameterize the noise level by $\xi := 1 - s \in (0, 1)$, where larger ξ means heavier noise.

DP traversal. Fig. 2a shows the empirical DP curves obtained by sweeping the lookahead t from s (MMSE) to 1 (full perception) across four noise levels $\xi \in \{0.1, 0.3, 0.5, 0.7\}$. As expected, the average denoiser traces smooth, monotone curves in the DP plane: increasing t improves perception (lower FID) at the cost of distortion (higher RMSE). The resulting curves exhibit the convex shape predicted by the optimal DP function, except at the highest noise level, where they become nearly linear; we attribute this to the flow-map approximation error over the long interval $[s, 1]$ (see Appendix C.9). Our approach thus yields a smooth and continuous parameterization of the DP tradeoff from a single trained model.

We also observe that the traversal range increases with the noise level. For $\xi = 0.1$, the range is narrow since the MMSE estimator is already close to the data distribution, whereas for $\xi = 0.7$, the lookahead spans a wide range of operating points. This is consistent with Freirich et al. (2021), as the perception gap P^* grows with noise, allowing a larger tradeoff between distortion and perception. Fig. 35 illustrates this effect: as t increases from s to 1, reconstructions transition from smooth averages to sharp, detailed images.

Algorithm 1 PnP with Flow Map Denoisers

Require: Trained flow map $\mathbf{v}_\theta(\cdot, s, t)$, observations $\mathbf{y}$, forward model f, iterations N, noise levels $s_k = k/N$, lookaheads $t_k \in [s_k, 1]$, step sizes $\{\lambda_k\}_{k=0}^{N-1}$

Initialize $\mathbf{x}^0$
for $k = 0, \dots, N-1$ **do**
  $\mathbf{z}^k = \mathbf{x}^k - \lambda_k \nabla g(\mathbf{x}^k)$ ▷ Gradient step (data fidelity)
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\tilde{\mathbf{z}}^k = (1 - s_k)\,\boldsymbol{\epsilon} + s_k\,\mathbf{z}^k$ ▷ Renoising to level s_k
  $\mathbf{x}^{k+1} = \tilde{\mathbf{z}}^k + (1 - s_k)\,\mathbf{v}_\theta(\tilde{\mathbf{z}}^k, s_k, t_k)$ ▷ Average denoiser $D_{s_k,t_k}(\tilde{\mathbf{z}}^k)$
end for
return $\mathbf{x}^N$

Variance restoration. A natural question is *why* increasing the lookahead improves perceptual quality. A partial explanation follows from the law of total variance: for any noisy observation $\mathbf{x}_s$, we have $\mathrm{Var}(\mathbf{x}_1) = \mathrm{Var}\big(\mathbb{E}[\mathbf{x}_1 \mid \mathbf{x}_s]\big) + \mathbb{E}\big[\mathrm{Var}(\mathbf{x}_1 \mid \mathbf{x}_s)\big]$. Since the second term is strictly positive, the MMSE estimator $D_s(\mathbf{x}_s) = \mathbb{E}[\mathbf{x}_1 \mid \mathbf{x}_s]$ has lower variance than the true data, i.e., $\mathrm{Var}(D_s(\mathbf{x}_s)) < \mathrm{Var}(\mathbf{x}_1)$. This *variance gap* leads to reduced contrast and over-smoothed reconstructions, and is precisely what the optimal DP estimator (7) corrects via interpolation. As the lookahead t increases, the average denoiser progressively restores this variance, effectively traversing the same tradeoff within a single model. Note that matching the variance alone is not sufficient, since adding independent noise also restores variance while degrading the reconstruction; for a Gaussian target, the optimal perfect-perception estimator is instead a deterministic transport map with no added noise (Freirich et al., 2021, Thm. 4 and Cor. 2), which $D_{s,1}$ matches. Fig. 2b supports this empirically: variance ratio and FID improve together, following a nearly identical relationship across noise levels, suggesting that variance restoration underlies the observed perceptual improvements.

4 Plug-and-play with average denoisers

We now embed the average denoiser (8) into the PnP framework from Section 2.2 to solve general inverse problems with continuous DP control. Importantly, the DP interpretation applies at the level of the denoiser $D_{s,t}$, while the overall PnP solution reflects its interaction with the forward model. As a result, in general inverse problems, the lookahead acts as a regularization parameter that controls a tradeoff combining perceptual alignment with data-consistency effects, rather than exactly reproducing the denoising DP frontier.

Algorithm. Given a trained flow map $\mathbf{v}_\theta(\cdot, s, t)$, observations $\mathbf{y}$, and a chosen lookahead, our method alternates between a gradient descent step on the data-fidelity term $g(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - f(\mathbf{x})\|^2$ and a proximal step utilizing the average denoiser. Following Martin et al. (2025), we include a renoising step prior to applying the denoiser. This reprojects the gradient-updated iterate onto the support of the distribution at noise level s_k, ensuring that the average denoiser D_{s_k,t_k} processes inputs at its intended operational noise level. The noise level is never tuned directly: it follows the Euler discretization $s_k = k/N$, $k = 0, \dots, N-1$, where N is the number of PnP iterations, as in prior PnP works with flow and diffusion priors. The lookahead is set with a constant offset Δ, i.e., $t_k = \min(s_k + \Delta, 1)$. The complete procedure is summarized in Algorithm 1.

Role of the renoising step. The stochastic renoising step is what makes the lookahead a DP knob in the inverse-problem setting. With renoising, $D_{s,t}$ operates at its intended noise level s, placing it in the regime where Theorem 1 characterizes its behavior. Importantly, the injected noise $(1 - s_k)\boldsymbol{\epsilon}$ is not a hyperparameter, so the only degree of freedom is the lookahead. Without renoising and for Gaussian priors, the iteration reduces to a fixed-point Tikhonov estimator and the lookahead degenerates into a bias-variance knob; we treat this case in Appendix D.5. In general, the iteration remains MAP-biased, with the lookahead changing the implicit regularizer, and converges due to the annealing schedule rather than contraction (Appendices D.5 and D.4).

Computational cost and connection to PnP-Flow. Algorithm 1 requires one forward pass through $\mathbf{v}_\theta$ per iteration, and reduces to PnP-Flow (Martin et al., 2025) for $t_k = s_k$. The lookahead t is merely

a conditioning input, adding zero overhead and preserving efficiency over methods requiring ODE backpropagation or trace computations.

5 Related work

Consistency models and flow maps for inverse problems. Consistency models (Song et al., 2023b) learn to map any point on a probability flow ODE to its origin, enabling single-step generation. Consistency trajectory models (Kim et al., 2025) and flow maps (Boffi et al., 2025a) generalize this to arbitrary pairs (s, t) by learning the ODE solution operator. More recent works repurposed these models for controlled generation, either as guided samplers (Sabour et al., 2025a; Spagnoletti et al., 2025; Garber and Tirer, 2025; Holderrieth et al., 2026), as priors (Gülle et al., 2026), or for optimizing the noise initialization (Mammadov et al., 2026). Most recently, and concurrently to this work, Huang et al. (2026) also leverage flow maps for guidance, formulating reward-based generation as an optimal control problem and using the flow map for fast guided sampling. However, all these approaches primarily target accelerated sampling at a fixed operating point and do not provide explicit control over the restoration tradeoff. In contrast, we treat the lookahead t as a continuous parameter, turning a single flow map into a family of estimators that traverses the distortion–perception (DP) plane.

Methods that traverse the DP plane. Recent approaches explore distortion–perception control by learning data-dependent flows between degraded and clean distributions. Inversion by Direct Iteration (InDI) (Delbracio and Milanfar, 2023) and stochastic interpolants (Albergo et al., 2024) learn flows from measurements to clean images, while Posterior-Mean Rectified Flow (PMRF) (Ohayon et al., 2025) replaces the source distribution with an approximation of the posterior mean, and OFTSR (Zhu et al., 2026) distills a one-step super-resolution flow with a tunable fidelity–realism parameter. However, they rely on supervised training with paired data $(\mathbf{y}, \mathbf{x})$ and, in the case of PMRF and OFTSR, require an auxiliary or teacher model. In contrast, our approach learns a single unconditional flow map from clean images alone, whose lookahead parameter naturally induces a continuous family of denoisers. As a result, DP control is obtained from a single model and extends naturally to arbitrary forward operators through Plug-and-Play inference.

Closer to our setting, several training-free methods traverse the DP plane with a single pre-trained model by acting on the sampler: DiffPIR (Zhu et al., 2023) scales the per-step injected noise, DDPG (Garber and Tirer, 2024) interpolates between back-projection and least-squares guidance, Wang et al. (2025) scale the injected noise of the reverse SDE, and MAP-RPS (Zhang et al., 2026) re-noises a MAP estimate before posterior sampling (see Appendix C.5 for further related methods). Where the knob acts determines which problem is solved. Because their knob controls the output per observation, these methods traverse the *conditional* DP plane, which requires a stochastic estimator. Our method acts on the denoiser with the solver fixed: instead of adding noise, we deterministically transport the estimator itself, traversing the *marginal* DP plane, which is what FID measures, and for which the optimal estimator is provably deterministic (Freirich et al., 2021). We compare with Wang et al. (2025) and MAP-RPS in Appendix C.5.

6 Numerical experiments

Throughout the experiments, we aim to (i) demonstrate competitive performance with state-of-the-art methods, and (ii) show that our method enables continuous traversal of the DP plane with a single model.

Setup. We evaluate all methods in two datasets: CelebA 128×128 and AFHQ 256×256. For each dataset, we train a flow map model $\mathbf{v}_\theta(\mathbf{x}, s, t)$, parametrized by the Song UNet architecture (Song et al., 2020; Ronneberger et al., 2015), from scratch using the LSD loss described in Section 2.1 (diagonal and off-diagonal terms). More training details can be found in Appendix B.1. We assess the reconstruction quality of the different samplers with mean squared error (in terms of $\mathrm{PSNR} = -10 \log_{10} \mathrm{MSE}$) and two perceptual metrics, LPIPS (Zhang et al., 2018) and DISTS (Ding et al., 2020). In addition, we compute the FID (Heusel et al., 2017) and KID (Bińkowski et al., 2018) between ensembles of test set images and posterior samples. We compute the average of 100 samples, and for all the methods we consider Gaussian noise with $\sigma = 0.05$, except for random inpainting with $\sigma = 0.01$; also, in Appendix C.2.2 we include an experiment with Poisson noise.

Table 1: Quantitative results for random inpainting (90% missing pixels, $\sigma = 0.01$) and motion deblurring ($\sigma = 0.05$) on CelebA and AFHQ. Rows are operating points on the DP plane, not a ranking: posterior samplers (e.g., DPS-ODE) target perception, optimization-based methods (e.g., D-Flow) target distortion. Best results are in **bold**; second best underlined.

Sampler	**CelebA** 128×128								**AFHQ** 256×256							
	Inpainting				Motion Deblurring				Inpainting				Motion Deblurring			
	PSNR↑	**LPIPS↓**	**FID↓**	**DISTS↓**	**PSNR↑**	**LPIPS↓**	**FID↓**	**DISTS↓**	**PSNR↑**	**LPIPS↓**	**FID↓**	**DISTS↓**	**PSNR↑**	**LPIPS↓**	**FID↓**	**DISTS↓**
Flow-priors	26.50	0.089	59.43	0.132	25.07	0.151	114.41	0.188	23.94	0.193	27.22	0.164	23.14	**0.207**	18.97	0.154
D-Flow	26.55	0.066	50.27	0.110	29.77	**0.071**	73.15	**0.112**	22.80	0.192	18.31	0.137	**25.52**	0.230	19.92	0.156
OT-ODE	22.16	0.158	84.92	0.179	21.07	0.198	106.78	0.202	21.14	0.263	16.03	0.181	16.73	0.348	77.53	0.234
DPS-ODE	25.60	0.070	**45.74**	**0.106**	27.36	0.101	57.02	0.133	23.08	0.127	**8.69**	0.098	20.49	0.292	36.71	0.179
PnP-Flow ($t = s$)	**27.29**	0.073	48.88	0.113	**29.79**	0.114	53.65	0.130	**24.99**	0.162	13.49	0.135	25.40	0.347	20.37	0.201
PnP-Flow ($t = 1$)	26.03	**0.065**	46.79	0.109	29.42	0.082	**52.61**	0.119	23.23	**0.122**	9.11	**0.095**	23.31	**0.207**	**14.04**	**0.137**

We compared with recent methods that leverage pre-trained flow models, namely OT-ODE (Pokle et al., 2024), Flow Priors (Zhang et al., 2024), D-Flow (Ben-Hamu et al., 2024), DPS-ODE (Chung et al., 2023) and PnP-Flow (Martin et al., 2025), which corresponds to our proposed method for $t = s$. For all the models, we performed a grid search of hyperparameters targeting the metric each method is designed for (perceptual metrics for posterior samplers, PSNR for optimization-based methods), and we use the same trained network for the prior/regularizer; we defer to Appendix B.2 for further details. We also include a comparison with baselines using a flow matching model as prior (a model trained only with the diagonal loss), which can be found in App. C.4.

6.1 Inverse problems

We evaluate on the following tasks: (i) motion deblurring with random kernel generated with the code[1] and size 31×31 (CelebA) and 61×61 (AFHQ) and $\sigma_b = 1$; (ii) random inpainting with 90% missing pixels; (iii) super-resolution ($\times 4$); and (iv) Gaussian deblurring, which are deferred to Appendix C.2.

Quantitative results are reported in Table 1 (with error bars in Appendix C.1), and visual results in Appendix C.10 and Fig. 4. PnP-Flow with $t = s$ achieves the best or near-best PSNR across all four (dataset, task) settings, confirming that the average denoiser at small lookaheads behaves as a low-distortion estimator. At the other extreme, PnP-Flow with $t = 1$ achieves the best or near-best LPIPS, DISTS, and FID on every task; on inpainting, it matches DPS-ODE on FID while substantially improving LPIPS and DISTS. Competing methods either commit to a single operating point (DPS-ODE favors perception, D-Flow favors distortion) or fall short on both, whereas our approach attains both regimes from a single trained network. Beyond accuracy, PnP-Flow is also the fastest of the methods compared, as shown in Appendix B.3. Lastly, we note that PnP-Flow with $t = 1$ is not uniformly the best perceptual method in all tasks. In particular, on Gaussian deblurring and super-resolution (in Appendix C.2), some baselines achieve better perceptual metrics. Nevertheless, increasing the lookahead from $t = s$ to $t = 1$ consistently shifts PnP-Flow toward the perceptual end of the DP spectrum, substantially improving FID, LPIPS, and DISTS relative to the distortion-oriented regime.

6.2 Analysis of the DP tradeoff

We now analyze how the lookahead parameter t controls the characteristics of our reconstructions by moving in the DP plane. We evaluate the average denoiser $\{D_{s,t}\}_t$ at multiple lookahead offsets Δ within a PnP solver and report RMSE (distortion) against FID (perception); all other hyperparameters are fixed per task and shared across lookaheads, and for box inpainting, we use a mask of size 80×80.

Results are in Fig. 3, and two observations stand out. First, varying t produces a smooth, (almost) monotonic trajectory in the DP plane: small lookaheads yield minimum RMSE, large lookaheads minimum FID, and intermediate values smoothly interpolate between the two. This is consistent with our analysis from Section 3.2.1 and suggests that the behavior extends to natural images and general degradations. Second, the same qualitative trend appears across all five inverse problems—Gaussian deblurring, motion deblurring, random/box inpainting, and super-resolution—despite the very different forward operators, suggesting that the DP control given by t is degradation-agnostic;

[1] `https://github.com/LeviBorodenko/motionblur`

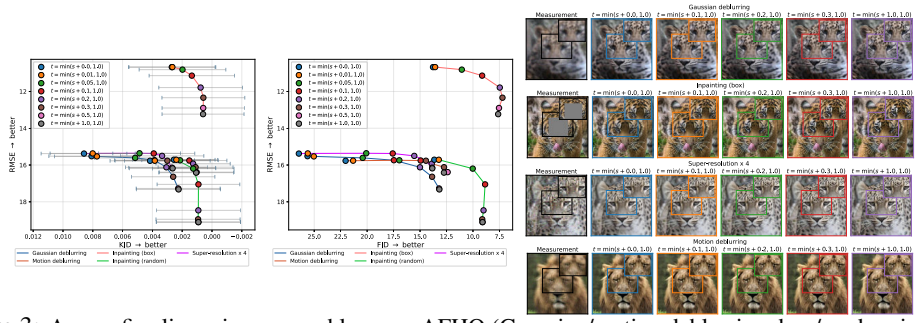
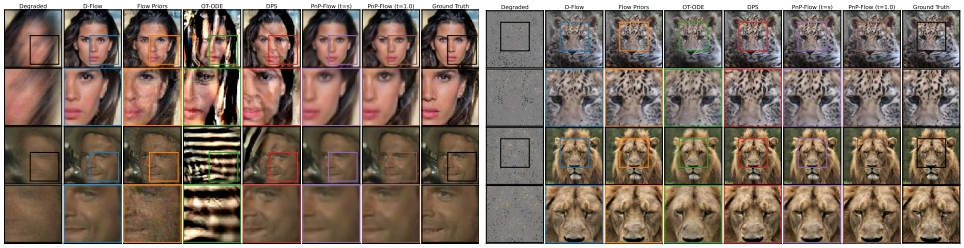


Figure 3: Across five linear inverse problems on AFHQ (Gaussian/motion deblurring, box/random inpainting, SR$\times$4), we sweep the lookahead t within a PnP solver and report RMSE (distortion) versus FID and KID (perception). Each marker corresponds to a lookahead offset Δ (with $t_k = \min(s_k + \Delta, 1)$), with small lookaheads favoring low distortion and large ones favoring low FID. A single trained model with a fixed solver configuration per task traverses the DP plane in all tasks; reconstructions on the right illustrate the perceptual change as t increases (larger version in Fig. 24).



(a) Motion deblurring (b) Random inpainting

Figure 4: Results on CelebA and AFHQ. PnP-Flow with $t = 1$ produces sharper reconstructions than with $t = s$, closer to those of posterior sampling methods.

the exception is box inpainting near $t = 1$, which we analyze in Appendix C.13. Finally, the same behavior holds for the non-linear JPEG restoration and colorization tasks (Appendix C.2.1).

7 Discussion and limitations

We introduced flow map denoisers, a generalization of the standard instantaneous denoiser that enables continuous, zero-shot traversal of the DP tradeoff. By reinterpreting flow maps as a continuum of average denoisers indexed by a lookahead parameter t, we link flow maps to the theory of optimal DP estimators. This traversal is provably optimal for isotropic Gaussian targets and empirically observed for natural images. Embedded within a Plug-and-Play (PnP) solver, our approach enables inference-time control over restoration quality across diverse degradations, without retraining, paired data, or auxiliary models.

Our approach has several limitations. First, the formal optimality result is restricted to the Gaussian case, and we do not provide convergence guarantees for the PnP iteration. In particular, certain degradations can lead to non-monotonic DP behavior, like box inpainting in Fig. 3, which we attribute to spatially coherent null spaces, and phase retrieval, which we attribute to the non-convex data fidelity (Appendix C.13). Second, flow map models are more expensive to train than standard flow matching models (about 10 versus 4 days on 8 A100 GPUs) and may underperform on pure distortion metrics such as PSNR. Third, while increasing the lookahead parameter consistently moves reconstructions toward the perceptual end of the DP spectrum, the resulting endpoint ($t = 1$) is not always the best perceptual method. This suggests that the lookahead parameter should primarily be viewed as a mechanism for traversing the DP tradeoff rather than as a way of reaching the optimal perceptual operating point for every restoration task.

Future directions include training average denoisers directly via SURE-based/consistency losses (Raphan and Simoncelli, 2011; Daras et al., 2025; Tachella et al., 2025). Lastly, our experiments are limited to pixel-space models (CelebA 128, AFHQ 256); extending the framework to latent-space and text-to-image settings is an important direction.

Acknowledgments and Disclosure of Funding

This work was supported in part by the NVIDIA Academic Grants Program. The research was sponsored by the National Science Foundation (CCF 2340481).

References

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. (2025). Stochastic interpolants: A unifying framework for flows and diffusions. *J. Mach. Learn. Res.*

Albergo, M. S., Goldstein, M., Boffi, N. M., Ranganath, R., and Vanden-Eijnden, E. (2024). Stochastic interpolants with data-dependent couplings. In *Intl. Conf. on Mach. Learn. (ICML)*, pages 921–937. PMLR.

Ben-Hamu, H., Puny, O., Gat, I., Karrer, B., Singer, U., and Lipman, Y. (2024). D-flow: Differentiating through flows for controlled generation. In *Intl. Conf. on Mach. Learn. (ICML)*. PMLR.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying MMD GANs. In *Intl. Conf. Learn. Repr. (ICLR)*.

Blau, Y. and Michaeli, T. (2018). The perception-distortion tradeoff. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*.

Boffi, N. M., Albergo, M. S., and Vanden-Eijnden, E. (2025a). Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Trans. Mach. Learn. Res (TMLR)*.

Boffi, N. M., Albergo, M. S., and Vanden-Eijnden, E. (2025b). How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Inf. Process. Syst. (NIPS)*.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. In *Intl. Conf. Learn. Repr. (ICLR)*.

Chung, H., Kim, J., and Ye, J. C. (2025). Diffusion models for inverse problems. *arXiv preprint arXiv:2508.01975*.

Daras, G., Chung, H., Lai, C.-H., Mitsufuji, Y., Ye, J. C., Milanfar, P., Dimakis, A. G., and Delbracio, M. (2024). A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*.

Daras, G., Rodriguez-Munoz, A., Klivans, A., Torralba, A., and Daskalakis, C. C. (2025). Ambient diffusion omni: Training good models with bad data. In *Advances in Neural Inf. Process. Syst. (NIPS)*.

Delbracio, M. and Milanfar, P. (2023). Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Trans. Mach. Learn. Res (TMLR)*.

Ding, K., Ma, K., Wang, S., and Simoncelli, E. P. (2020). Image quality assessment: Unifying structure and texture similarity. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(5):2567–2581.

Dong, C., Loy, C. C., He, K., and Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):295–307.

Flamary, R., Courty, N., et al. (2021). Pot: Python optimal transport. *J. Mach. Learn. Res.*, 22(78):1–8.

Freirich, D., Michaeli, T., and Meir, R. (2021). A theory of the distortion-perception tradeoff in wasserstein space. *Advances in Neural Inf. Process. Syst. (NIPS)*, 34:25661–25672.

Garber, T. and Tirer, T. (2024). Image restoration by denoising diffusion models with iteratively preconditioned guidance. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 25245–25254.

Garber, T. and Tirer, T. (2025). Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 2398–2407.

Geng, Z., Deng, M., Bai, X., Kolter, J. Z., and He, K. (2025). Mean flows for one-step generative modeling. In *Advances in Neural Inf. Process. Syst. (NIPS)*.

Gribonval, R. and Nikolova, M. (2021). On Bayesian estimation and proximity operators. *Applied and Computational Harmonic Analysis*, 50:49–72.

Gülle, M., Yun, J., Alçalar, Y. U., and Akçakaya, M. (2026). PnP-CM: Consistency models as plug-and-play priors for inverse problems. *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Inf. Process. Syst. (NIPS)*, 30.

Holderrieth, P., Chen, D., Eyring, L., Shah, I., Anantharaman, G., He, Y., Akata, Z., Jaakkola, T., Boffi, N. M., and Simchowitz, M. (2026). Diamond maps: Efficient reward alignment via stochastic flow maps. *arXiv preprint arXiv:2602.05993*.

Hu, Y., Delbracio, M., Milanfar, P., and Kamilov, U. S. (2024). A restoration network as an implicit prior. In *Intl. Conf. Learn. Repr. (ICLR)*.

Huang, J. Y., Lin, J., Shah, S., Nair, K., and Boffi, N. M. (2026). How to guide your flow: Few-step alignment via flow map reward guidance. *arXiv preprint arXiv:2604.27147*.

Hurault, S., Leclaire, A., and Papadakis, N. (2022). Gradient step denoiser for convergent plug-and-play. In *Intl. Conf. Learn. Repr. (ICLR)*.

Kadkhodaie, Z. and Simoncelli, E. (2021). Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Inf. Process. Syst. (NIPS)*, 34:13242–13254.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Advances in Neural Inf. Process. Syst. (NIPS)*, 35:26565–26577.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. (2024). Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 24174–24184.

Kawar, B., Vaksman, G., and Elad, M. (2021). SNIPS: Solving noisy inverse problems stochastically. *Advances in Neural Inf. Process. Syst. (NIPS)*, 34:21757–21769.

Kim, B., Kim, J., Kim, J., and Ye, J. C. (2025). Generalized consistency trajectory models for image manipulation. In *Intl. Conf. Learn. Repr. (ICLR)*.

Kingma, D. P. (2015). Adam: A method for stochastic optimization. *Intl. Conf. Learn. Repr. (ICLR)*.

Laumont, R., Bortoli, V. D., Almansa, A., Delon, J., Durmus, A., and Pereyra, M. (2022). Bayesian imaging using plug & play priors: when Langevin meets Tweedie. *SIAM J. Imag. Sciences*, 15(2):701–737.

Laumont, R., De Bortoli, V., Almansa, A., Delon, J., Durmus, A., and Pereyra, M. (2023). On maximum a posteriori estimation with plug & play priors and stochastic gradient descent. *J. Math. Imaging Vis.*, 65(1):140–163.

Lee, C., Yoo, J., Agarwal, M., Shah, S., Huang, J., Raghunathan, A., Hong, S., Boffi, N. M., and Kim, J. (2026). One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *Intl. Conf. Learn. Repr. (ICLR)*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*.

Lu, Y., Lu, S., Sun, Q., Zhao, H., Jiang, Z., Wang, X., Li, T., Geng, Z., and He, K. (2026). One-step latent-free image generation with pixel mean flows. *Intl. Conf. on Mach. Learn. (ICML)*.

Mammadov, A., Takao, S., Chen, B., Baptista, R., Mardani, M., Teh, Y. W., and Berner, J. (2026). Variational flow maps: Make some noise for one-step conditional generation. *arXiv preprint arXiv:2603.07276*.

Mardani, M., Song, J., Kautz, J., and Vahdat, A. (2024). A variational perspective on solving inverse problems with diffusion models. *Intl. Conf. Learn. Repr. (ICLR)*.

Martin, S., Gagneux, A., Hagemann, P., and Steidl, G. (2025). PnP-flow: Plug-and-play image restoration with flow matching. In *Intl. Conf. Learn. Repr. (ICLR)*.

Miyasawa, K. et al. (1961). An empirical Bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist*, 38(181-188):1–2.

Ohayon, G., Adrai, T., Vaksman, G., Elad, M., and Milanfar, P. (2021). High perceptual quality image denoising with a posterior sampling CGAN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1805–1813.

Ohayon, G., Michaeli, T., and Elad, M. (2025). Posterior-mean rectified flow: Towards minimum mse photo-realistic image restoration. In *Intl. Conf. Learn. Repr. (ICLR)*.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. (2020). Deep learning techniques for inverse problems in imaging. *IEEE J.Sel. Areas Inf. Theory.*, 1(1):39–56.

Pesquet, J.-C., Repetti, A., Terris, M., and Wiaux, Y. (2021). Learning maximally monotone operators for image recovery. *SIAM Journal on Imaging Sciences*, 14(3):1206–1237.

Pokle, A., Muckley, M. J., Chen, R. T. Q., and Karrer, B. (2024). Training-free linear image inverses via flows. *Trans. Mach. Learn. Res (TMLR)*.

Raphan, M. and Simoncelli, E. P. (2011). Least squares estimation without priors or supervision. *Neural Comp.*, 23(2):374–420.

Renaud, M., Prost, J., Leclaire, A., and Papadakis, N. (2024). Plug-and-play image restoration with stochastic denoising regularization. In *Intl. Conf. on Mach. Learn. (ICML)*.

Robbins, H. E. (1956). An empirical Bayes approach to statistics. In *Proc. Third Berkeley Symp. Math. Statist. Prob.*, pages 388–394. Springer.

Romano, Y., Elad, M., and Milanfar, P. (2017). The little engine that could: Regularization by denoising (RED). *SIAM J. Imag. Sciences*, 10(4):1804–1844.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, pages 234–241. Springer.

Sabour, A., Albergo, M. S., Domingo-Enrich, C., Boffi, N. M., Fidler, S., Kreis, K., and Vanden-Eijnden, E. (2025a). Test-time scaling of diffusions with flow maps. *arXiv preprint arXiv:2511.22688*.

Sabour, A., Fidler, S., and Kreis, K. (2025b). Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Inf. Process. Syst. (NIPS)*.

Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. In *Intl. Conf. Learn. Repr. (ICLR)*.

Song, J., Vahdat, A., Mardani, M., and Kautz, J. (2022). Pseudoinverse-guided diffusion models for inverse problems. In *Intl. Conf. Learn. Repr. (ICLR)*.

Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.-Y., Kautz, J., Chen, Y., and Vahdat, A. (2023a). Loss-guided diffusion models for plug-and-play controllable generation. In *Intl. Conf. on Mach. Learn. (ICML)*, pages 32483–32498. PMLR.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. (2023b). Consistency models. In *Intl. Conf. on Mach. Learn. (ICML)*, pages 32211–32252. PMLR.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *Intl. Conf. Learn. Repr. (ICLR)*.

Spagnoletti, A., Prost, J., Almansa, A., Papadakis, N., and Pereyra, M. (2025). Latino-pro: Latent consistency inverse solver with prompt optimization. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 19597–19607.

Tachella, J., Davies, M. E., and Jacques, L. (2025). Unsure: self-supervised learning with unknown noise level and stein's unbiased risk estimate. In *Intl. Conf. Learn. Repr. (ICLR)*.

Terris, M., Hurault, S., Song, M., and Tachella, J. (2026). Reconstruct anything model: a lightweight foundation model for computational imaging. *Intl. Conf. Learn. Repr. (ICLR)*.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. In *IEEE Global Conf. Signal and Info. Process. (GlobalSIP)*, pages 945–948. IEEE.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674.

Wang, Y., Bi, S., and Zhang, Y.-J. A. (2026). Training-free rate-distortion-perception traversal with diffusion. In *Intl. Conf. on Mach. Learn. (ICML)*.

Wang, Y., Bi, S., Zhang, Y.-J. A., and Yuan, X. (2025). Traversing distortion-perception tradeoff using a single score-based generative model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2377–2386.

Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. (2023). Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Inf. Process. Syst. (NIPS)*.

Zhang, J., Liu, Z., Yan, L., Xiao, Z., and Gu, Y. (2026). Stage-wise distortion-perception traversal in zero-shot inverse problems with diffusion models. In *Intl. Conf. on Mach. Learn. (ICML)*.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 586–595.

Zhang, Y., Yu, P., Zhu, Y., Chang, Y., Gao, F., Wu, Y. N., and Leong, O. (2024). Flow priors for linear inverse problems via iterative corrupted trajectory matching. *Advances in Neural Inf. Process. Syst. (NIPS)*, 37:57389–57417.

Zheng, H., Chu, W., Zhang, B., Wu, Z., Wang, A., Feng, B., Zou, C., Sun, Y., Kovachki, N. B., Ross, Z. E., et al. (2025). Inversebench: Benchmarking plug-and-play diffusion priors for inverse problems in physical sciences. In *Intl. Conf. Learn. Repr. (ICLR)*.

Zhu, Y., Wang, R., Lu, S., Li, J., Yan, H., and Zhang, K. (2026). OFTSR: One-step flow for image super-resolution with tunable fidelity-realism trade-offs. In *Intl. Conf. Learn. Repr. (ICLR)*.

Zhu, Y., Zhang, K., Liang, J., Cao, J., Wen, B., Timofte, R., and Van Gool, L. (2023). Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 1219–1229.

Zilberstein, N., Mardani, M., and Segarra, S. (2025). Repulsive latent score distillation for solving inverse problems. In *Intl. Conf. Learn. Repr. (ICLR)*.

Appendix

A Background

A.1 Flow Maps

Consider a probability flow defined by the ordinary differential equation

$$\dot{\mathbf{x}}_t = \mathbf{v}_t(\mathbf{x}_t), \quad \mathbf{x}_0 \sim p_0,$$

which transports samples from a simple distribution p_0 to a target distribution p_1. The associated *flow map* $\mathbf{X}_{s,t} : \mathbb{R}^d \to \mathbb{R}^d$ is defined as the solution operator of this ODE:

$$X_{s,t}(\mathbf{x}_s) = \mathbf{x}_t,$$

i.e., it maps a point at time s to its position at time t along the flow trajectory. In particular, $X_{0,1}$ directly maps samples from p_0 to p_1, enabling one-step generation without numerical integration. A convenient parameterization of the flow map is given by

$$X_{s,t}(\mathbf{x}) = \mathbf{x} + (t-s)\,\mathbf{v}(\mathbf{x}, s, t),$$

where $\mathbf{v}_{s,t}$ can be interpreted as the average velocity between times s and t. In the limit $t \to s$, this recovers the instantaneous velocity field:

$$\lim_{s \to t} \partial_t X_{s,t}(\mathbf{x}) = \mathbf{v}(\mathbf{x}, t),$$

which implies that $\mathbf{v}(\mathbf{x}, t, t) = \mathbf{v}(\mathbf{x}, t)$.

Characterization of the flow map. The flow map can be equivalently characterized in three ways:

- **Lagrangian form:**
$$\partial_t X_{s,t}(\mathbf{x}) = \mathbf{v}_t(X_{s,t}(\mathbf{x})),$$
which describes evolution along trajectories;
- **Eulerian form:**
$$\partial_s X_{s,t}(\mathbf{x}) + \nabla X_{s,t}(\mathbf{x})\,\mathbf{v}_s(\mathbf{x}) = 0,$$
which expresses conservation along the flow;
- **Semigroup property:**
$$X_{s,t}(\mathbf{x}) = X_{u,t}(X_{s,u}(\mathbf{x})),$$
which encodes consistency across time intervals.

These equivalent formulations provide the basis for learning flow maps in practice (Boffi et al., 2025b).

Self-distillation framework. Instead of learning the instantaneous velocity field $\mathbf{v}(., t)$ and integrating the ODE, flow map models directly learn $\mathbf{v}(., s, t)$ by combining:

- the *diagonal loss* in (2) enforcing $\mathbf{v}(\mathbf{x}, t, t) \approx \mathbf{v}(\mathbf{x}, t)$,
- an *off-diagonal loss* enforcing consistency with one of the above three characterizations.

Notice that throughout this work we used the LSD loss in (5), which is the one associated to the Lagrangian condition. This approach, known as *self-distillation*, allows training a single model that approximates the full flow map $X_{s,t}$ without requiring a pre-trained teacher model.

A.2 Inverse problems with flow/diffusion priors

While this work focuses on flow-based models, a large body of work uses diffusion priors, which are closely related to our method. We briefly summarize some of these works.

These methods generate a sample from the posterior by running the reverse process using conditional score at t obtained via Bayes' rule as

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t). \tag{12}$$

While the second term uses a pre-trained diffusion model, the first is intractable, as seen from $p(\mathbf{y}|\mathbf{x}_t) = \int p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t)\mathrm{d}\mathbf{x}_0$. Prior works (Chung et al., 2023; Song et al., 2022; Kadkhodaie and Simoncelli, 2021; Song et al., 2023a) address this with a Gaussian approximation of $p(\mathbf{x}_0|\mathbf{x}_t)$ using Tweedie's formula $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \frac{1}{\alpha_t}\left(\mathbf{x}_t - \sigma_t \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right)$. This requires computing the *score Jacobian*, which is computationally expensive, especially for pixel-space models at high resolution. This approximation is the one used by DPS-ODE in Section 6.

Beyond guidance and PnP/optimization-based techniques, another line of work uses sequential Monte Carlo (SMC) (Wu et al., 2023), whose particle-based approximations target the exact posterior in the limit of infinitely many particles.

B Implementation

B.1 Training details

All models are trained unconditionally with the self-distillation objective of Boffi et al. (2025b): the diagonal loss (2) on $\mathbf{v}_\theta(\cdot, t, t)$ plus the off-diagonal LSD term (5). We also compared the off-diagonal objectives of Boffi et al. (2025b) (LSD, PSD, and ESD) and selected LSD, which trained stably and gave the best visual quality, consistent with their analysis.

Architecture. We use the SongUNet (Song et al., 2021), a UNet with self-attention, with a base channel width of 128 and channel multipliers $[1, 2, 3, 4]$, giving four resolution scales. Self-attention is applied at spatial resolutions 16×16 and 8×8. Each resolution level contains 3 residual blocks, dropout rate 0.1, and a cosine resampling filter $[1, 3, 3, 1]$.

Time conditioning. The network is conditioned on two scalars $(\alpha_r, \alpha_t) \in [0, 1]^2$ representing the interpolation times of the source and target. Both are encoded with a *positional (sinusoidal) embedding*. Following Boffi et al. (2025b), the two conditioning scalars are embedded as $\mathrm{PE}(\alpha_t - \alpha_r)$ and $\mathrm{PE}(\alpha_r)$, summed, and passed through a two-layer MLP with SiLU activations to produce a global conditioning vector injected into every residual block via adaptive group normalization. Compared with a flow matching model of the same architecture, the only addition is this second time embedding, so both models have approximately 128M parameters.

Loss weighting. For the off-diagonal term, we sample s, t uniformly from $\{(s, t) : 0 < s < t \leq 1\}$ as detailed in Boffi et al. (2025b). Following Karras et al. (2024), we learn a per-sample uncertainty weight for the loss: a small network takes the pair (α_r, α_t) and outputs a scalar log-variance $\log \sigma^2(\alpha_r, \alpha_t) \in \mathbb{R}$, which weights the loss as $\mathcal{L} \propto e^{-\log \sigma^2} \ell + \log \sigma^2$. This adaptive weighting automatically rescales the diagonal and off-diagonal terms to comparable magnitudes, so no manual balancing is required.

Datasets and resolution.

- **CelebA** 128×128 (Liu et al., 2015): 202,599 images; 90% train split.
- **AFHQ** 256×256 (Choi et al., 2020): 16,130 images; 90% train split.

During training, 75% of each batch is used for the diagonal loss and the remaining 25% for the off-diagonal one. We adopted this split from Boffi et al. (2025b) and did not tune it; with it, training was stable on both datasets, without divergence, restarts, or per-dataset tuning. Augmentation (horizontal flip, rotation, scaling, anisotropic scaling, translation) is applied with probability $p = 0.12$.

Optimization. All models are optimized with Adam (Kingma, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\varepsilon = 10^{-8}$) at a learning rate of 2×10^{-4} with no learning-rate schedule.

Compute. For training, we use 8 NVIDIA A100 GPUs, while for inference we use a single GPU for all methods. Flow maps are more expensive to train than flow matching models, because the off-diagonal term requires a Jacobian–vector product at every step. On CelebA, reaching the reported quality took about 10 days on 8 A100 GPUs (with good results after 6–7 days), versus about 4 days for a flow matching model with the same architecture and data (the model of Appendix C.4). No large-scale pre-trained flow map is currently available, so we trained all models from scratch; our

construction only requires evaluating a flow map at pairs (s, t) with $s < t$, and thus applies to any such model once available.

Table 2: Per-run training configuration.

Dataset	Resolution	Batch size	GPUs	Days
CelebA	128×128	128	8	10 ($\approx$ 700k iterations)
AFHQ	256×256	32	8	9 ($\approx$ 580k iterations)

B.2 Hyperparameters and baselines

For all baselines, unless otherwise specified, we use $\sigma = 0.05$ for Gaussian/motion deblurring and super-resolution, and $\sigma = 0.01$ for inpainting. The blur kernel size is set to 61 for AFHQ and 31 for CelebA. For each method, hyperparameters are grid-searched around the values suggested by the authors, targeting the metric the method is designed to optimize: perceptual metrics for posterior samplers (DPS-ODE, OT-ODE) and PSNR for optimization-based methods (Flow-Priors, D-Flow, PnP-Flow with $t = s$).

Flow-Priors (Zhang et al., 2024). We follow the original parameterization with guidance weight λ and step size η. We use $\lambda = 10^5$, $\eta = 10^{-2}$, and $K = 1$ inner iteration in all settings. The number of outer iterations is fixed to $N = 100$.

D-Flow (Ben-Hamu et al., 2024). We tune the regularization parameter $\lambda \in \{0.1, 0.01, 0.001\}$ and the initialization blending parameter $\alpha \in \{0.1, 0.3, 0.5\}$, retaining $\lambda = 0.01$ and $\alpha = 0.1$. As recommended by the authors, the inner LBFGS optimization runs for 20 iterations and the ODE is solved with 6 Euler steps. The number of outer iterations is fixed to 20.

OT-ODE (Pokle et al., 2024). We tune the initial time $t_0 \in \{0.1, 0.2, 0.3, 0.4\}$ and the step-size schedule $\gamma \in \{\text{constant}, \sqrt{t}\}$. We use $\gamma = \text{constant}$ throughout. The retained values are $t_0 = 0.1$ for inpainting, $t_0 = 0.3$ for Gaussian deblurring and super-resolution, and $t_0 = 0.4$ for motion deblurring on AFHQ. The number of iterations is fixed to $N = 100$.

DPS-ODE (Chung et al., 2023). This method corresponds to the ODE (deterministic) version of DPS, originally designed for diffusion models. We use the ODE version to match the flow matching literature (Zhang et al., 2024); DPS is often reported to perform better with stochasticity, and stochastic samplers are covered by the comparison of Appendix C.5. We tune the guidance weight $\eta \in \{10^2, 10^3, 10^4\}$, retaining $\eta = 10^3$ as default, $\eta = 10^2$ on CelebA box inpainting, and $\eta = 10^2$ on AFHQ Gaussian deblurring. The number of iterations is fixed to $N = 100$.

PnP-Flow w/ $t = s$ (Martin et al., 2025) and w/ $t = 1$ (ours). We adjust the learning-rate exponent $\alpha \in \{0.05, 0.1\}$ and the number of iterations $N \in \{30, 50, 100, 300\}$, which also sets the noise schedule $s_k = k/N$. We use $\alpha = 0.05$ in most settings and $\alpha = 0.1$ for motion deblurring; $N = 100$ by default, $N = 300$ for random inpainting on AFHQ, and $N \in \{30, 50\}$ for motion deblurring.

B.3 Computational cost

Running time. Table 3 reports per-image reconstruction times on a single A100 GPU. PnP-Flow is the fastest method we compare against, while reaching perceptual quality typically associated with posterior sampling. This illustrates a concrete advantage of using a flow map as a PnP denoiser: it inherits the speed of MAP-style approaches without sacrificing perceptual fidelity.

Table 3: Running time for each method. PnP with a flow map as a denoiser is the fastest method, while generating images that are closer to those from posterior sampling methods.

Sampler	Time [sec]
Flow-priors	34.8
D-Flow	322
OT-ODE	8.4
DPS-ODE	7.7
PnP-flow	4.5

Memory footprint. Overall, almost all the methods fit in a single A100 with a batch of 5 images for AFHQ 256x256; the only one that could not fit is D-flow, which needed to use a single image per batch.

B.4 Experiment of the teaser plot

We detail here the experiment underlying Fig. 1. The data distribution p_1 is a 2D isotropic mixture of Gaussians,

$$p_1 = \tfrac{1}{2}\mathcal{N}(\boldsymbol{\mu}_a, \sigma_c^2\mathbf{I}) + \tfrac{1}{2}\mathcal{N}(\boldsymbol{\mu}_b, \sigma_c^2\mathbf{I}), \quad \boldsymbol{\mu}_a = (4.5, 3.5),\ \boldsymbol{\mu}_b = (3.5, 5.5),\ \sigma_c = 0.35.$$

Observations are generated according to the linear forward model in (6), with $f(\cdot) = \mathbf{H}$ a fixed rotation of angle $\pi/6$ followed by isotropic rescaling.

Similarly to the case in Appendix D.3, the MMSE denoiser $D_s(x) = \mathbb{E}[X_1 \mid X_s = x]$ admits a closed-form expression via the component posteriors. We use this analytic denoiser to construct the flow-map denoiser

$$D_{s,t}(\mathbf{x}) = \mathbf{x} + (1-s)\,\mathbf{v}(\mathbf{x}, s, t), \quad \mathbf{v}(\mathbf{x}, s, t), = \frac{1}{t-s}\int_s^t \frac{D_\tau(\mathbf{x}_\tau) - \mathbf{x}_\tau}{1-\tau}\, d\tau,$$

where the average velocity is approximated by Euler integration of the flow $\dot{\mathbf{x}}_\tau = (D_\tau(\mathbf{x}_\tau) - \mathbf{x}_\tau)/(1-\tau)$ using 100 steps. Using this approximation of the average denoiser, we run PnP-Flow in Alg. 1 with $s = 0.45$ (which is fixed to isolate the behavior of the lookahead), step size $\gamma_k = \lambda\sqrt{1-k/K}$ with $\lambda = 0.4$, and $K = 50$ iterations. We initialize with $\mathbf{x}^{(0)} = \mathbf{H}^{-1}\mathbf{y}$ and run the algorithm independently for each sample. We consider three lookaheads $t \in \{s,\ s+0.27,\ 1\}$, corresponding to the MMSE, an intermediate, and the full lookahead regime.

The behavior follows the distortion–perception tradeoff: at $t = s$, reconstructions collapse toward the inter-mode mean (low variance, low distortion), while at $t = 1$ the bimodal structure is recovered (higher perceptual quality). Intermediate lookaheads interpolate smoothly between these regimes, all using a single denoiser.

Trajectories induced by the flow map. Fig. 5 shows, for a subset of samples, how the lookahead moves each reconstruction. Let $\tilde{\mathbf{z}}^K$ be the input of the denoiser at the last iteration; we plot the path $\tau \mapsto D_{s,\tau}(\tilde{\mathbf{z}}^K)$ for $\tau \in [s, t]$. It starts at the MMSE estimate $D_{s,s}(\tilde{\mathbf{z}}^K)$ and ends at the PnP output $D_{s,t}(\tilde{\mathbf{z}}^K)$. At $t = s$ the path reduces to a point between the two modes; as t increases, the flow map transports the estimate away from the posterior mean and into one of the modes, which restores the bimodal structure of p_1.

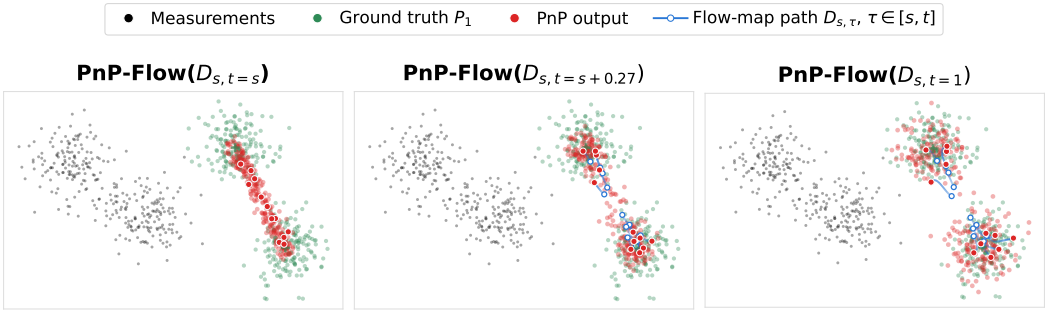


Figure 5: Same experiment as Fig. 1, with the paths $\tau \mapsto D_{s,\tau}(\tilde{\mathbf{z}}^K)$, $\tau \in [s,t]$, induced by the flow map for 12 samples (blue; open circle: MMSE estimate at $\tau = s$; red: PnP output at $\tau = t$).

C Additional experiments

C.1 Results with error bars

Here, we provide the results from Section 6 with the corresponding error bars.

Table 4: Quantitative results for random inpainting (90%) with $\sigma = 0.01$ on CelebA and AFHQ. Best results are in **bold**.

Sampler	CelebA 128×128				AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓	PSNR↑	LPIPS↓	FID↓	DISTS↓
Flow-priors	26.50 ± 1.00	0.089 ± 0.011	59.43	0.132 ± 0.012	23.94 ± 0.79	0.193 ± 0.021	18.96	0.164 ± 0.011
D-Flow	26.55 ± 2.79	0.066 ± 0.031	50.27	0.110 ± 0.031	22.80 ± 3.80	0.192 ± 0.146	18.31	0.137 ± 0.083
OT-ODE	22.16 ± 0.74	0.158 ± 0.019	84.92	0.179 ± 0.016	21.14 ± 0.26	0.263 ± 0.019	16.03	0.181 ± 0.012
DPS-ODE	25.60 ± 1.32	0.070 ± 0.054	**45.74**	**0.106** ± 0.024	23.08 ± 0.73	0.127 ± 0.021	**8.69**	0.098 ± 0.014
PnP-Flow ($t = s$)	**27.29** ± 1.15	0.073 ± 0.073	48.88	0.113 ± 0.013	**24.99** ± 0.78	0.162 ± 0.013	13.49	0.135 ± 0.008
PnP-Flow ($t = 1$)	26.03 ± 1.03	**0.065** ± 0.009	46.79	0.109 ± 0.012	23.23 ± 0.739	**0.122** ± 0.012	9.11	**0.095** ± 0.006

Table 5: Quantitative results for motion deblurring with $\sigma = 0.05$ on CelebA and AFHQ. Best results are in **bold**.

Sampler	CelebA 128×128				AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓	PSNR↑	LPIPS↓	FID↓	DISTS↓
Flow-priors	25.07 ± 1.34	0.151 ± .035	114.41	0.188 ± 0.021	23.14 ± 2.26	**0.207** ± 0.063	27.73	0.154 ± 0.028
D-Flow	**29.77** ± 1.67	**0.071** ± 0.022	73.15	**0.112** ± 0.019	**25.52** ± 2.60	0.230 ± 0.085	19.92	0.156 ± 0.042
OT-ODE	21.07 ± 7.43	0.198 ± 0.157	106.78	0.202 ± 0.113	16.73 ± 5.51	0.348 ± 0.165	77.53	0.234 ± 0.122
DPS-ODE	27.36 ± 5.16	0.101 ± 0.063	57.02	0.133 ± 0.054	20.49 ± 3.66	0.292 ± 0.123	36.71	0.179 ± 0.60
PnP-Flow ($t = s$)	29.79 ± 2.70	0.114 ± 0.056	53.65	0.130 ± 0.034	25.40 ± 2.15	0.347 ± 0.076	20.37	0.201 ± 0.033
PnP-Flow ($t = 1$)	29.42 ± 2.80	0.082 ± 0.042	**52.61**	0.119 ± 0.029	23.31 ± 2.50	**0.207** ± 0.059	**14.04**	**0.137** ± 0.028

C.2 Additional inverse problems

Gaussian deblurring. We consider a kernel of size 31×31 for CelebA and 61×61 for AFHQ, with $\sigma_b = 3$. The quantitative results are in Table 6.

Super-resolution. We consider super-resolution with a cubic method, and a factor of 4. The quantitative results are in Table 7.

Box-inpainting. We consider a box of size 80×80 for AFHQ and 60×60 for CelebA. The quantitative results are in Table 8.

Table 6: Quantitative results for Gaussian deblurring w/$\sigma = 0.05$ across CelebA and AFHQ datasets. Best results are in **bold**.

Sampler	CelebA 128×128				AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓	PSNR↑	LPIPS↓	FID↓	DISTS↓
Flow-priors	**30.41**	**0.044**	80.53	0.125	22.73	0.182	12.87	0.124
D-Flow	26.59	0.081	66.93	**0.120**	22.80	0.175	12.31	0.121
OT-ODE	25.77	0.100	63.10	0.129	24.04	**0.168**	11.51	**0.114**
DPS-ODE	24.80	0.129	72.13	0.148	20.62	0.253	15.47	0.158
PnP-Flow ($t = s$)	27.25	0.233	70.80	0.183	**25.14**	0.453	25.67	0.235
PnP-Flow ($t = 1$)	26.60	0.157	**61.57**	0.152	23.41	0.191	**11.09**	0.122

Table 7: Quantitative results for super-resolution $\times 4$ w/$\sigma = 0.05$ across CelebA and AFHQ datasets. Best results are in **bold**.

Sampler	CelebA 128×128				AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓	PSNR↑	LPIPS↓	FID↓	DISTS↓
Flow-priors	26.14	0.227	96.46	0.211	22.93	0.338	20.41	0.209
D-Flow	24.84	0.137	119.93	0.181	23.77	0.389	20.69	0.204
OT-ODE	24.79	0.142	81.18	0.173	20.36	0.400	44.28	0.247
DPS-ODE	26.38	**0.083**	**55.10**	**0.123**	24.18	**0.165**	**10.88**	**0.114**
PnP-Flow ($t = s$)	**26.94**	0.235	71.30	0.187	**25.10**	0.429	26.45	0.232
PnP-Flow ($t = 1$)	26.68	0.166	63.97	0.158	24.71	0.269	13.88	0.159

Table 8: Quantitative results for box inpainting w/$\sigma = 0.01$ across CelebA and AFHQ datasets. Best results are in **bold**.

Sampler	CelebA 128×128				AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓	PSNR↑	LPIPS↓	FID↓	DISTS↓
Flow-priors	25.09	0.062	36.28	0.068	26.51	0.048	16.62	0.052
D-Flow	26.36	**0.056**	34.70	0.073	26.87	0.069	11.08	0.079
OT-ODE	24.65	0.065	31.99	0.073	20.80	0.085	25.97	0.095
DPS-ODE	24.13	0.115	58.64	0.132	21.07	0.069	19.48	0.071
PnP-Flow ($t = s$)	**26.77**	0.059	32.29	**0.062**	**28.33**	0.057	13.79	0.052
PnP-Flow ($t = 1$)	24.77	0.062	**26.07**	**0.062**	26.40	**0.031**	**7.61**	**0.033**

C.2.1 Non-linear

JPEG. We consider JPEG restoration with quantization factor 10. Since OT-ODE and Flow-Priors cannot handle non-linear degradations, we only compare with DPS-ODE; the results on AFHQ are in Table 9. Fig. 6 shows the effect of the number of PnP iterations: the perceptual endpoint consistently achieves better perceptual quality, although increasing the number of iterations eventually degrades FID, as in the linear problems (Appendix C.12).

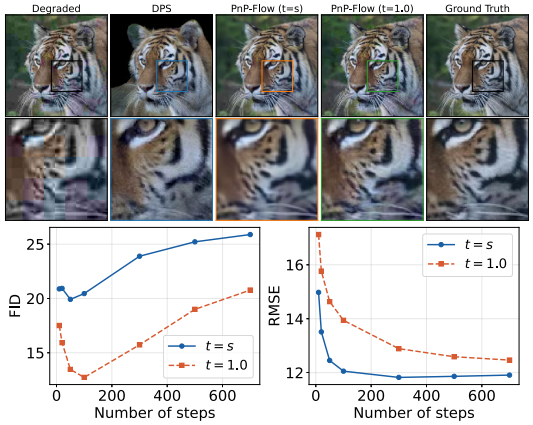


Figure 6: DP behavior on JPEG restoration (AFHQ) as a function of the number of PnP iterations.

Table 9: Quantitative results for JPEG w/$\sigma = 0.01$ on AFHQ datasets. Best results are in **bold**.

Sampler	AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓
DPS-ODE	18.76	0.287	19.05	0.184
PnP-Flow ($t = s$)	**26.79**	0.245	20.33	0.166
PnP-Flow ($t = 1$)	25.52	**0.143**	**12.84**	**0.124**

Colorization. We consider also Colorization; the result is shown in Table 10 for AFHQ.

Table 10: Quantitative results for Colorization w/$\sigma = 0.001$ on AFHQ datasets. Best results are in **bold**.

Sampler	AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓
DPS-ODE	18.60	0.413	33.79	0.236
PnP-Flow ($t = s$)	**26.30**	0.111	9.95	0.110
PnP-Flow ($t = 1$)	25.37	**0.105**	**9.47**	**0.094**

C.2.2 Gaussian deblurring with Poisson noise

Lastly, we consider Poisson noise, to show that the method is not limited to Gaussian measurement noise. We normalize each image to $[0, 1]$ and consider independent Poisson observations normalized by a peak rate $\lambda > 0$:

$$y_i \,|\, \mathbf{x} \;\sim\; \text{Poisson}\big(\lambda\,(\mathbf{H}\mathbf{x})_i\big), \qquad i = 1, \ldots, N, \tag{13}$$

so that $\mathbb{E}[y_i \mid \mathbf{x}] = (\mathbf{H}\mathbf{x})_i$ and $\text{Var}[y_i \mid \mathbf{x}] = (\mathbf{H}\mathbf{x})_i/\lambda$. The signal-to-noise ratio scales as $\sqrt{\lambda}$, where lower λ corresponds to fewer photons and a noisier observation; we consider $\lambda = 20$. Given this noise model, the corresponding negative log-likelihood, used as the data-fidelity term in Alg. 1, is (up to a constant in $\mathbf{x}$)

$$g(\mathbf{x}) \;=\; \sum_{i=1}^{N} \big[\,(\mathbf{H}\mathbf{x})_i \;-\; y_i \log(\mathbf{H}\mathbf{x})_i\,\big]. \tag{14}$$

The results as a function of the number of steps is shown in Fig. 7, while qualitative comparisons are in Fig. 22.

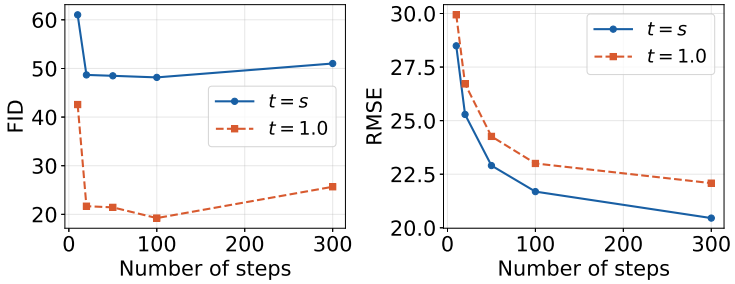


Figure 7: Ablation of PnP-flow as a function of the number of steps with Poisson noise and Gaussian deblurring

C.3 Distortion-perception of CelebA

Fig. 8 reproduces the experiment of Section 6.2 on CelebA. Again, varying t produces a smooth, nearly monotonic trajectory in the DP plane: small lookaheads minimize RMSE, large lookaheads minimize FID, and intermediate values interpolate between the two.

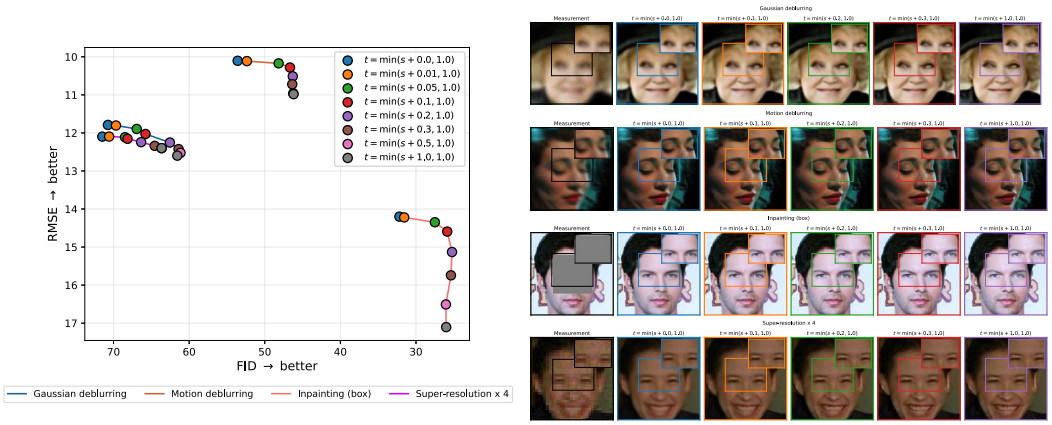


Figure 8: **Distortion-perception tradeoff traced by a single flow map on CelebA.** Same setup as Fig. 3, but on CelebA 128×128. For each of four linear inverse problems (Gaussian deblurring, motion deblurring, box inpainting, and super-resolution), we sweep the lookahead t of our average denoiser within a PnP solver and report RMSE (distortion, $\downarrow$) against FID (perception, $\downarrow$). Each marker corresponds to a different value of t; small lookaheads cluster near the low-distortion regime (top of each curve) while large lookaheads cluster near the low-FID regime (right). The same trained model traverses the DP plane in all four tasks. Reconstructions on the right illustrate the perceptual change as t increases.

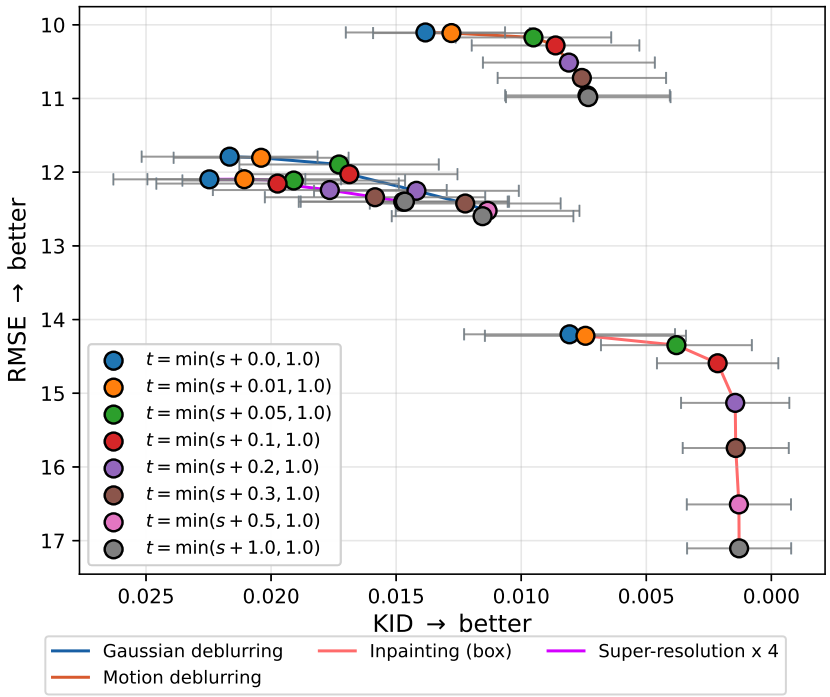


Figure 9: **Distortion-perception tradeoff traced by a single flow map on CelebA.** Same setup as Fig. 8, reporting KID instead of FID.

C.4 Comparison with baselines using a flow matching model

Table 11 compares all methods under both a flow matching backbone and a flow map backbone, allowing us to isolate the contribution of the flow map training objective from the choice of sampler.

Flow matching backbone. PnP-Flow ($t = s$) achieves the highest PSNR (25.36 dB), but at the cost of perceptual quality relative to Flow-priors (LPIPS 0.359 vs. 0.205). PnP-Flow ($t = 1$) is inapplicable here, as it requires the consistency property enforced only by flow map training.

Impact of the backbone. The results are largely consistent across backbones, with DPS-ODE achieving a lower FID with the flow matching model. However, the flow map backbone with PnP-flow ($t = 1$) still yields notably lower FID scores throughout and is required to unlock PnP-flow ($t = 1$).

Table 11: Quantitative results for motion deblurring w/$\sigma = 0.05$ across AFHQ dataset. Best results are in **bold**.

Sampler	Flow Matching backbone				Flow Map backbone			
	PSNR↑	LPIPS↓	FID↓	DISTS↓	PSNR↑	LPIPS↓	FID↓	DISTS↓
Flow-priors	23.20	0.205	18.48	<u>0.153</u>	<u>23.14</u>	<u>0.207</u>	18.96	<u>0.154</u>
D-Flow					25.52	0.230	<u>19.92</u>	0.156
OT-ODE	17.2	0.335	68.71	0.224	17.73	0.348	77.53	0.234
DPS-ODE	20.62	0.262	25.71	0.162	20.49	0.292	36.71	0.179
PnP-Flow ($t = s$)	25.36	0.359	20.90	0.206	**25.40**	0.347	20.37	<u>0.201</u>
PnP-Flow ($t = 1$)	-	-	-	-	23.31	**0.207**	**14.04**	**0.137**

C.5 Comparison with training-free DP-traversal methods

We compare our lookahead sweep with two training-free methods that traverse the DP plane with a single pre-trained model: Wang et al. (2025), which scales the noise injected in the reverse SDE by a parameter λ, and MAP-RPS (Zhang et al., 2026), which sets the time t_0 at which a re-noised posterior-sampling stage starts from a MAP initialization.

Setup. Since no public code is available for Wang et al. (2025), we implemented their sampler; for MAP-RPS, we adapted the official code. Both baselines share the same backbone: a flow matching model with the same architecture, data, and training pipeline as our flow map, converted into a score model through the affine relation between velocity and score for Gaussian paths. Their hyperparameters are tuned starting from the configurations reported in the respective papers. For our method, all solver hyperparameters are fixed and only the lookahead offset Δ varies; we use $N = 50$ iterations for Gaussian deblurring and $N = 300$ for random inpainting. All methods are evaluated on the same 50 test images.

Table 12: Comparison with training-free DP-traversal methods on AFHQ 256×256, at three values of each method's knob. RMSE and FID are both lower-is-better.

		Gaussian deblurring ($\sigma = 0.05$)		Random inpainting ($\sigma = 0.01$)	
Method	Knob values	RMSE↓	FID↓	RMSE↓	FID↓
Wang et al. (2025)	$\lambda \in \{0, 0.5, 1\}$	23.85 / 21.84 / 23.24	138.78 / 49.57 / 27.52	39.58 / 17.46 / 18.12	357.35 / 43.59 / 25.75
MAP-RPS	$t_0 \in \{0, 50, 100\}$	15.36 / 15.48 / 15.86	41.80 / 31.18 / 26.80	21.44 / 19.31 / 18.77	55.52 / 32.82 / 27.33
Ours	$t \in \{s, s{+}0.4, 1\}$	15.94 / 17.61 / 18.03	37.83 / 24.63 / 24.48	15.47 / 18.91 / 18.94	27.84 / 24.22 / 24.23

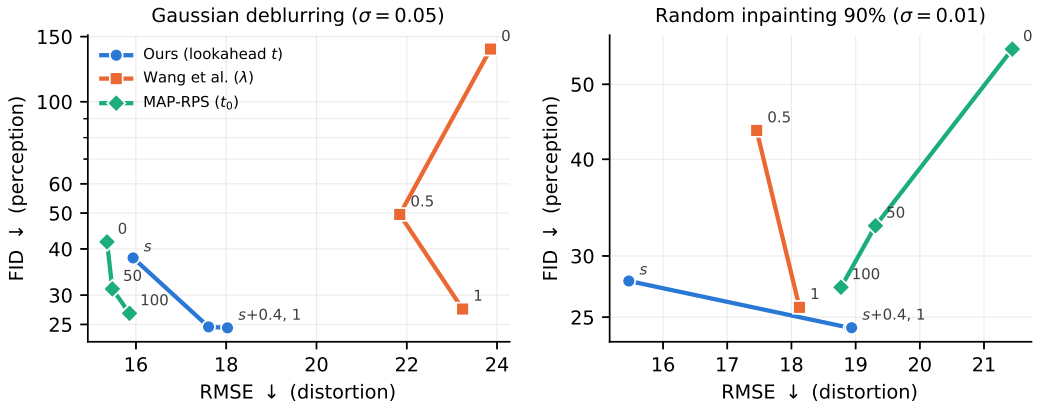


Figure 10: Operating points of Table 12 in the RMSE–FID plane (FID on a log scale), labeled by each method's knob. For random inpainting, Wang et al. (2025) at $\lambda = 0$ lies outside the axes (RMSE 39.58, FID 357.35).

Results. Table 12 and Fig. 10 report the results. On Gaussian deblurring, our operating points Pareto-dominate those of Wang et al. (2025). On random inpainting, their $\lambda = 0$ endpoint collapses (FID 357; the deterministic reverse mean fails on the mask, and we could not recover it by tuning), and their sweep does not reach the low-distortion end (best RMSE 17.46 versus 15.47 for ours). Compared with MAP-RPS, the curves cross on Gaussian deblurring: MAP-RPS attains lower distortion and we attain lower FID. On random inpainting, our curve lies below and to the left of theirs, i.e., it is better on both axes. In terms of runtime, processing a batch of 5 images takes 12 s for our method, 40 s for Wang et al. (2025), and 60 s for MAP-RPS, since both baselines backpropagate through the network at every step. Beyond performance, the two families differ in where the knob acts. The baselines modify the sampler and retune it per operating point (e.g., Wang et al. (2025) retune their guidance step per λ and per task, and MAP-RPS adjusts its sampling stage per t_0), whereas our sweep uses a single solver configuration for all t and changes only the conditioning of the denoiser (see also Section 5).

Other DP-traversal methods. Wang et al. (2026) extend the approach of Wang et al. (2025) to rate–distortion–perception traversal with a score-scaled probability-flow ODE, and prove Gaussian optimality for the conditional problem, complementary to our Theorem 1. The posterior-sampling conditional GAN of Ohayon et al. (2021) is a natural baseline, but no model is available for AFHQ or CelebA. Both of its mechanisms (scaling the latent noise and averaging samples) traverse between a posterior sampler and its conditional mean, placing it in the same conditional family as the two baselines above, both of which include it as a baseline in their experiments. OFTSR (Zhu et al., 2026), InDI (Delbracio and Milanfar, 2023), and PMRF (Ohayon et al., 2025) are trained on paired data for a specific degradation, whereas our unconditional prior is reused across operators without retraining; a fair comparison would require training a task-specific model for each of our settings, which we leave for future work.

C.6 Endpoint interpolation versus lookahead

Appendix D.2.3 shows that, for Gaussian targets, the convex combination (29) of the two endpoint denoisers traces the optimal DP curve exactly, even for anisotropic covariances. For images and inverse problems, however, this certificate no longer applies: its optimality relies on $D_{s,1}$ being the minimum-distortion perfect-perception estimator, i.e., on the coupling induced by the flow-map ODE being the optimal transport coupling, which need not hold in general. The two constructions then trace different paths between the same endpoints: the convex combination follows the pixel-space segment between the endpoint outputs, while the intermediate-t evaluation is a nonlinear network output. Moreover, the convex combination requires two network evaluations per iteration.

We compare both constructions within the same PnP solver on AFHQ Gaussian deblurring (200 images, $\sigma = 0.05$), replacing the denoiser in Algorithm 1 by $(1-\lambda)D_{s,1} + \lambda D_{s,s}$ for the interpolation (Fig. 11). At the same parameter value ($\lambda = \Delta = 0.5$), the two constructions land at very different operating points: the lookahead is already deep in the perceptual regime (KID 1.68×10^{-3}, RMSE

17.26), whereas the interpolation remains near the distortion end (KID 6.78×10^{-3}, RMSE 15.51); matching the latter requires $\Delta \approx 0.1$. At matched distortion (RMSE 15.54 versus 15.51), the lookahead attains a better KID (3.64 versus 6.78×10^{-3}; LPIPS 0.354 versus 0.394) at half the runtime (≈ 8.3 versus ≈ 16.4 s per batch), and its curve lies below the interpolation curve at every intermediate operating point, for both metrics. The lookahead therefore parametrizes the perceptual regime much faster than the interpolation, at no loss in quality and half the cost. A systematic comparison of both constructions is an interesting direction for future work.

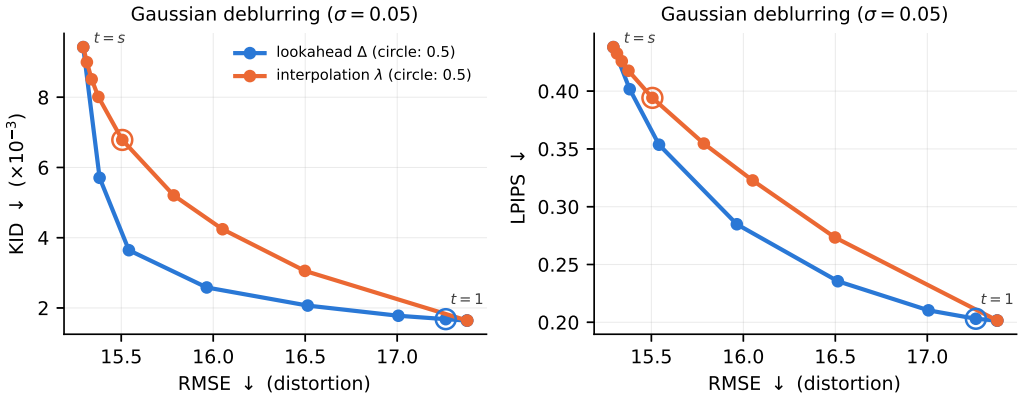


Figure 11: Endpoint interpolation versus lookahead within the same PnP solver (AFHQ Gaussian deblurring, 200 images; KID against the AFHQ test set), with $\lambda \in \{0, 0.1, 0.2, 0.3, 0.5, 0.7, 0.8, 0.9, 1\}$ and $\Delta \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 1\}$. Both curves share the endpoints $t = s$ and $t = 1$; circles mark $\lambda = \Delta = 0.5$.

C.7 Renoising magnitude

In Algorithm 1, the injected noise $(1 - s_k)\boldsymbol{\epsilon}$ has a single valid magnitude: it reprojects the iterate onto the interpolant level s_k at which the denoiser is trained. To verify that it cannot replace the lookahead as a DP knob, we scale it by a factor c, i.e., $\tilde{\mathbf{z}}^k = c\,(1 - s_k)\,\boldsymbol{\epsilon} + s_k\,\mathbf{z}^k$, with $\Delta = 0$ and all other hyperparameters fixed, on AFHQ Gaussian deblurring and random inpainting (200 images, same solver settings as in Appendix C.5). Fig. 12 compares the resulting operating points with the lookahead sweep. Reducing the noise ($c < 1$) degrades perceptual quality on both tasks without a meaningful gain in distortion (e.g., at $c = 0.75$, KID 11.04 versus 9.43×10^{-3} on deblurring, and 4.93 versus 2.96×10^{-3} on inpainting). Increasing it ($c > 1$) moves deblurring toward the perceptual regime, but every operating point is dominated by the lookahead sweep (e.g., at $c = 1.5$, RMSE 17.21 and KID 3.10×10^{-3}, versus 17.26 and 1.68×10^{-3} for $\Delta = 0.5$), and the iteration degrades on both axes at $c = 2$ (RMSE 22.40). On inpainting, $c > 1$ breaks the reconstruction (RMSE 28.90 at $c = 1.25$ and 67.15 at $c = 1.5$). LPIPS leads to the same conclusions (Fig. 12, bottom). The renoising magnitude therefore does not provide a useful traversal, and the lookahead remains the only degree of freedom.

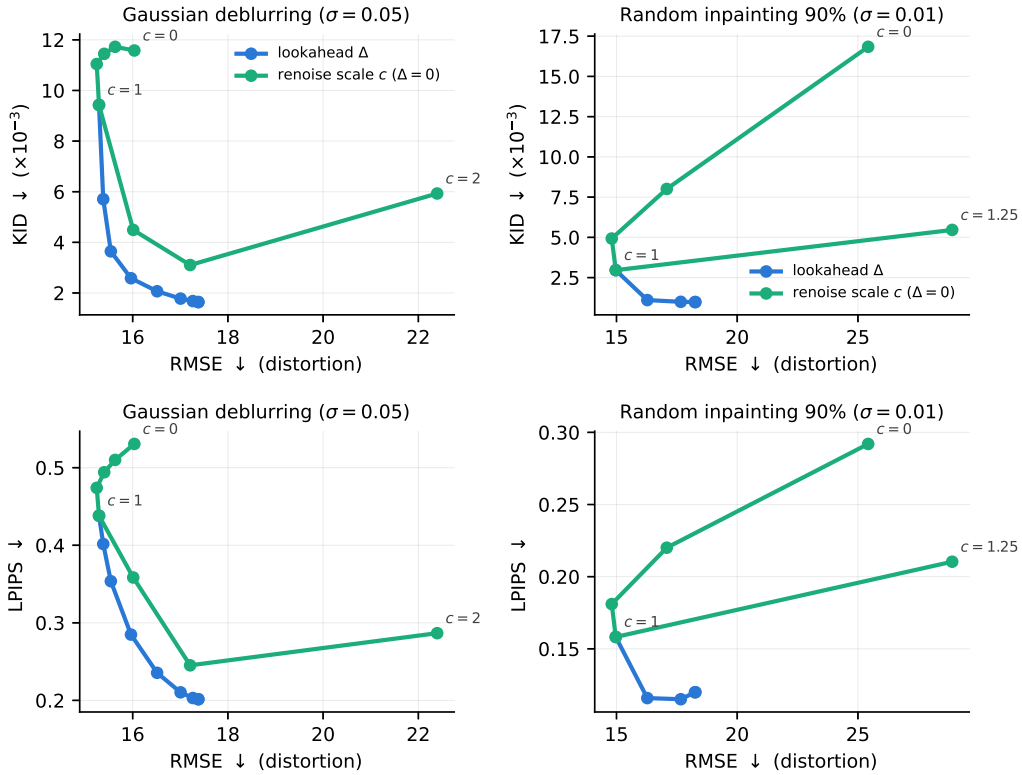


Figure 12: Scaling the injected noise by $c \in \{0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2\}$ at $\Delta = 0$ (deblurring; $c \in \{0, 0.5, 0.75, 1, 1.25, 1.5\}$ for inpainting), compared with the lookahead sweep (200 images). Top: KID; bottom: LPIPS. $c = 1$ coincides with $\Delta = 0$. For inpainting, $c = 1.5$ lies outside the axes (RMSE 67.15, KID 130.4×10^{-3}, LPIPS 0.589).

C.8 Measurement consistency along the lookahead

A perceptual gain obtained by drifting away from the measurements would show up as a growing residual. We measure $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}(t)\|$ along the lookahead sweep (200 images per setting), as a root mean square over the observed entries, normalized by the noise level σ (Fig. 13). For Gaussian deblurring and random inpainting, the residual does not depend on t: it stays at the noise level for deblurring ($1.00\,\sigma$) and below it for random inpainting (0.41–$0.43\,\sigma$). For the block masks of Appendix C.13, it increases slightly for small lookaheads and saturates by $\Delta \approx 0.3$; the largest increase ($0.23\,\sigma$, for 1×1 blocks) corresponds to about 0.3 gray levels on the observed pixels, whereas the RMSE increases by 4.3 gray levels over the same sweep. The perceptual gains therefore do not come from weakened data consistency.

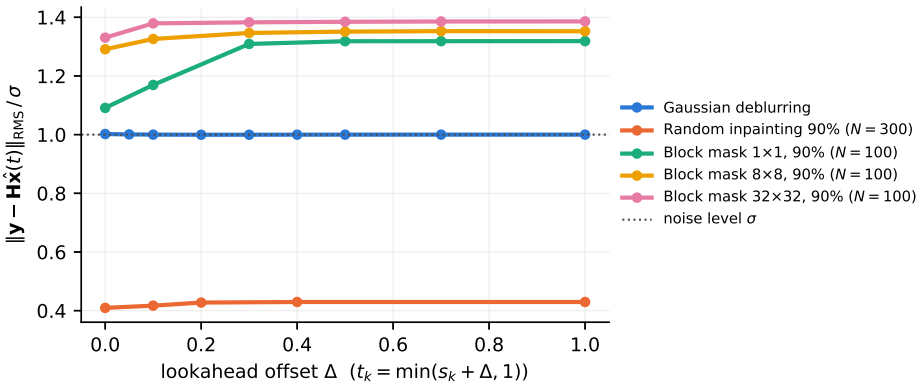


Figure 13: Measurement residual along the lookahead sweep on AFHQ, normalized by the noise level σ (RMS over observed entries). The residual of the clean image equals $1.00\,\sigma$ in all settings.

C.9 Convexity of the denoising DP curves and flow-map accuracy

In Fig. 2a, the DP curves are convex at low noise and nearly linear at the highest noise level. Since the convexity of the ideal DP curve does not depend on the noise level, we attribute this to a deviation of the learned model rather than to the DP characterization: reaching the perceptual endpoint requires traversing the long interval $[s, 1]$, where the single-step flow map is least accurate, so the deviation should grow at large ξ (small s) and depend on the accuracy of the flow map.

To test this, we evaluate the average denoiser with $K \in \{1, 2, 4, 8\}$ flow-map steps over $[s, t]$ and measure (i) the FID at the perceptual endpoint and (ii) the convexity of the DP curve, quantified by the normalized signed area between the curve and the chord joining its endpoints (positive for convex, negative for concave). Results are in Table 13.

Table 13: Convexity of the denoising DP curve on CelebA (normalized signed area; positive means convex) and FID at the perceptual endpoint, as a function of the number K of flow-map steps used to evaluate the average denoiser.

	Signed area				Endpoint FID	
ξ	$K = 1$	$K = 2$	$K = 4$	$K = 8$	$K = 1$	$K = 8$
0.1	$+0.23$	$+0.17$	$+0.18$	$+0.20$	16.3	16.8
0.3	$+0.08$	$+0.05$	$+0.15$	$+0.14$	36.9	34.1
0.5	-0.06	$+0.06$	-0.00	-0.04	52.0	46.1
0.7	-0.17	-0.11	-0.15	-0.17	68.4	57.8

Additional flow-map steps improve the endpoint FID at moderate and high noise (e.g., at $\xi = 0.7$, FID decreases from 68.4 to 62.7, 59.9, and 57.8 for $K = 1, 2, 4, 8$), which confirms the single-step map as a source of approximation error; at $\xi = 0.1$, the endpoint is nearly saturated and the change is within evaluation noise. The added accuracy restores convexity at moderate noise ($\xi = 0.3$ becomes clearly convex), but a residual concavity remains at the largest noise level that does not diminish with K. We attribute it to the error of the trained flow map in the heavy-noise regime, rather than to single-step discretization or to the DP characterization itself.

C.10 Visual results

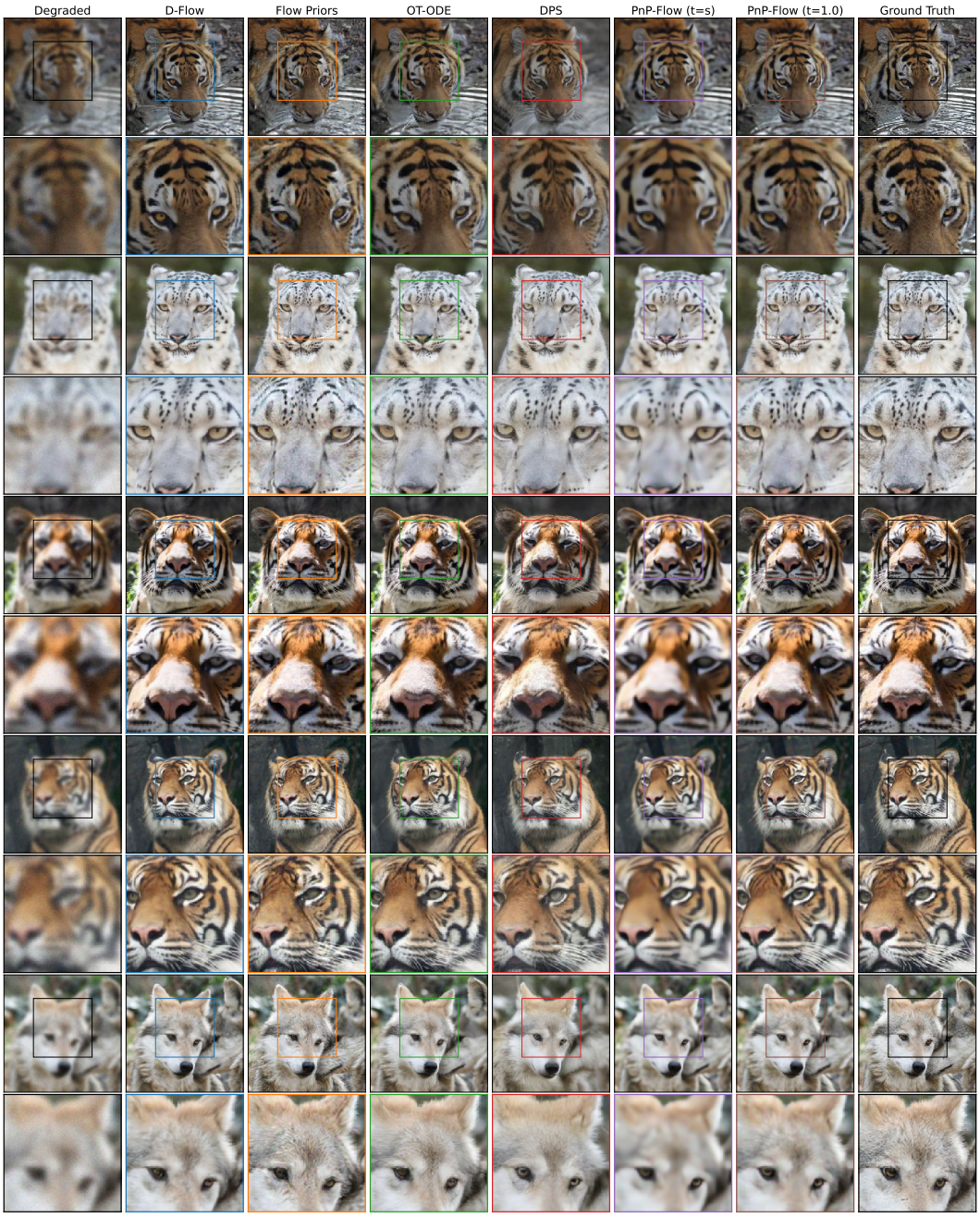


Figure 14: Gaussian deblurring with kernel size 61×61 and $\sigma_b = 3$, and $\sigma = 0.05$

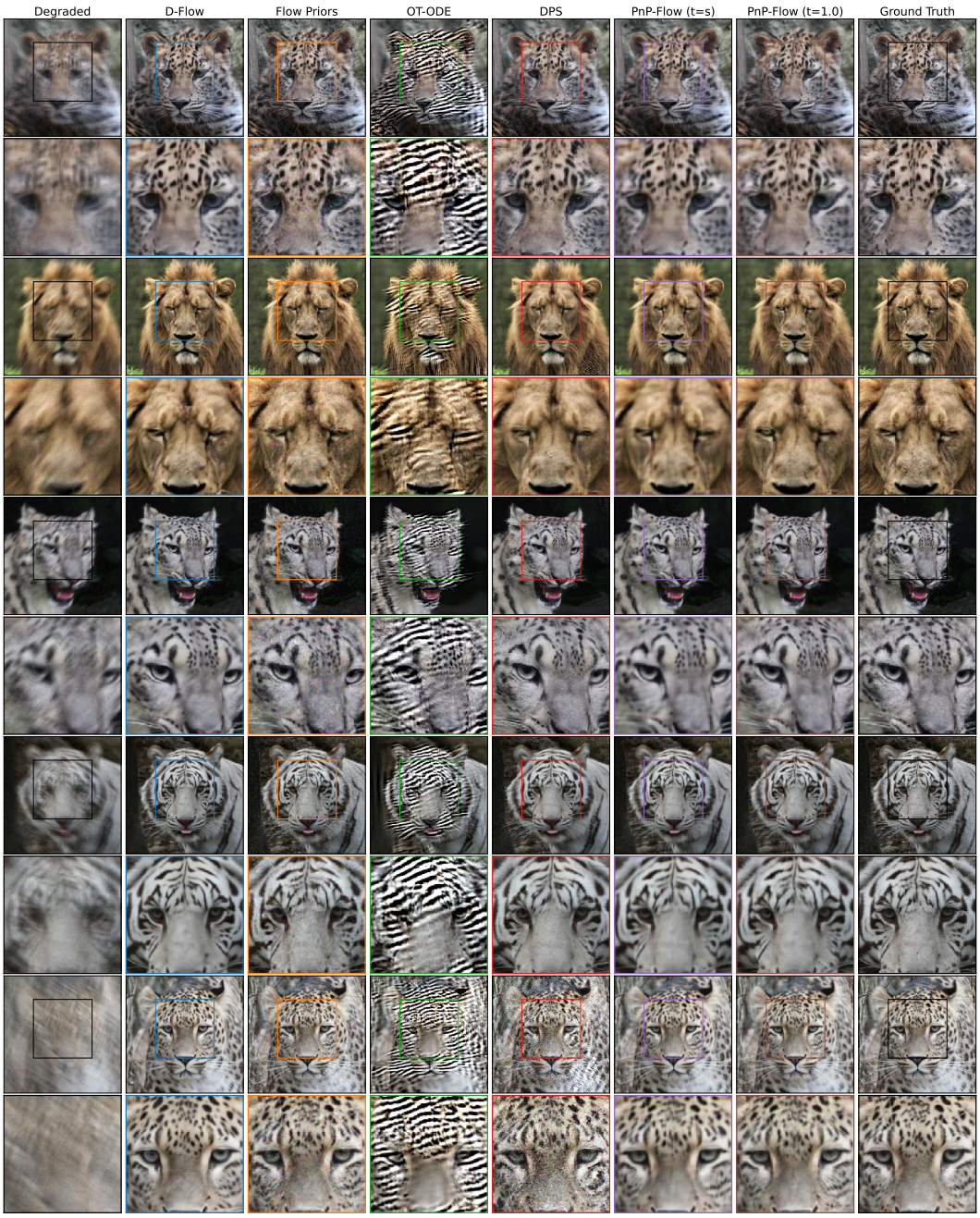


Figure 15: Motion deblurring with kernel size 61×61 and $\sigma_b = 1$, and $\sigma = 0.1$

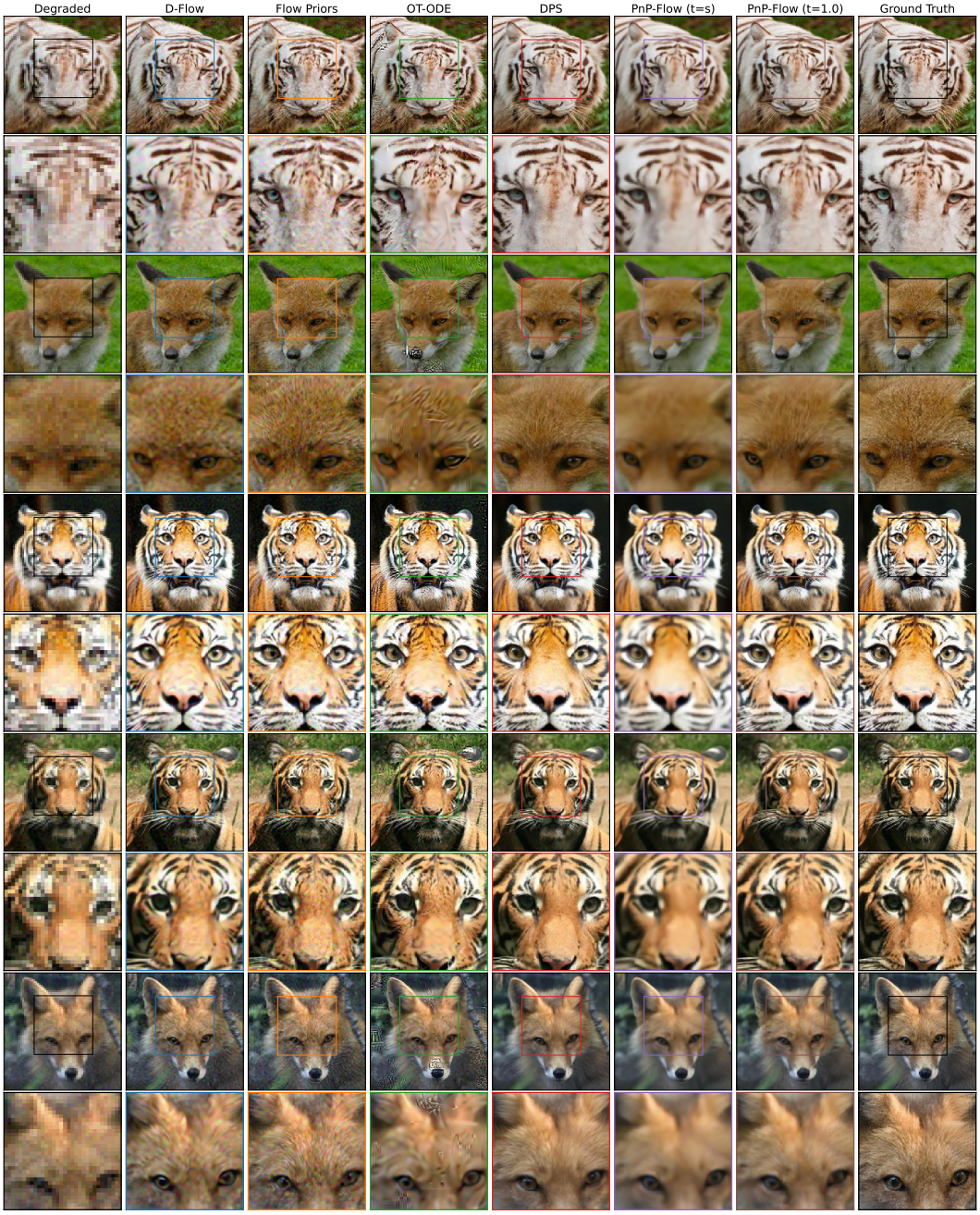


Figure 16: Super resolution $\times 4$, and $\sigma = 0.05$

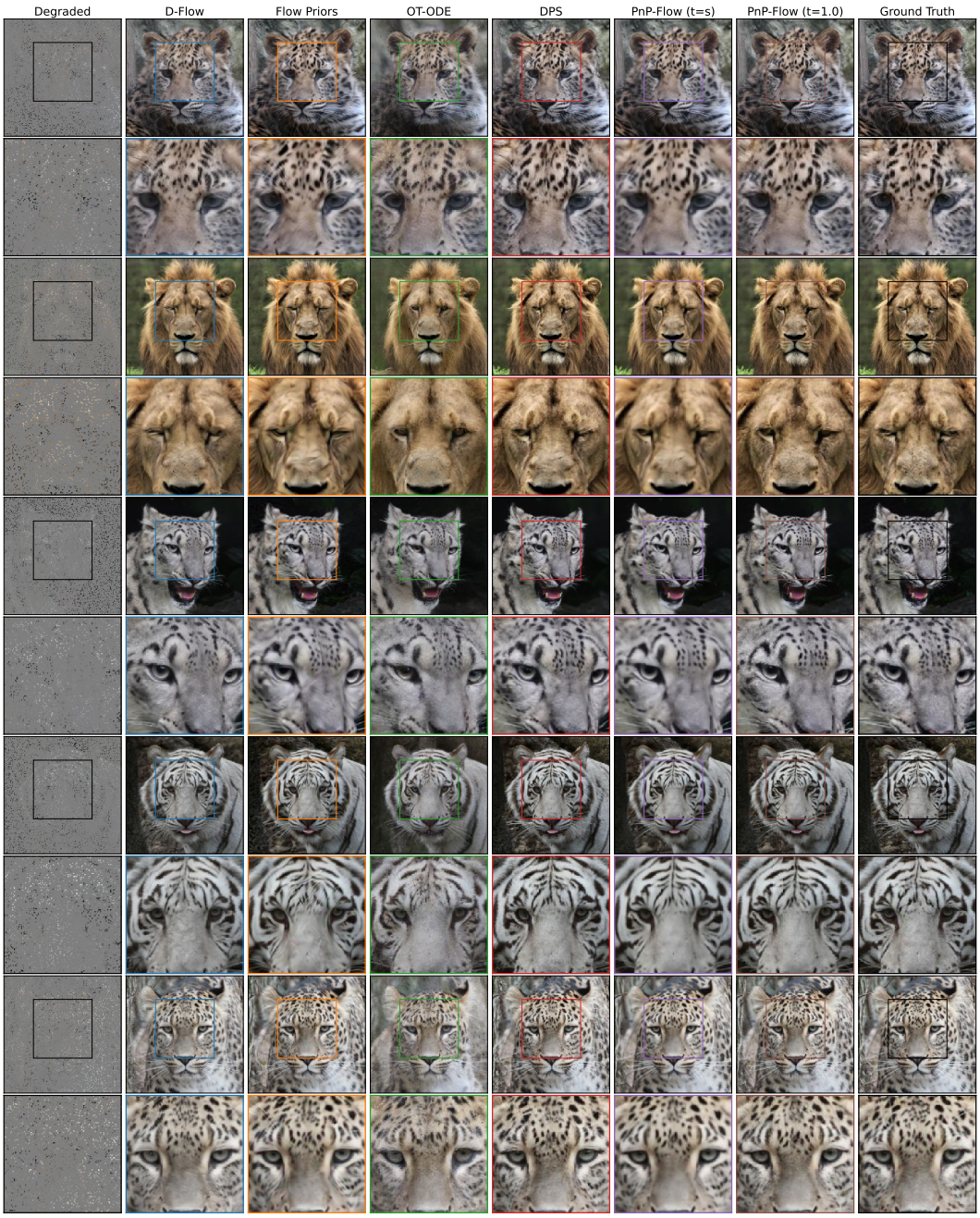


Figure 17: Random inpainting with 90% drop rate, and $\sigma = 0.01$

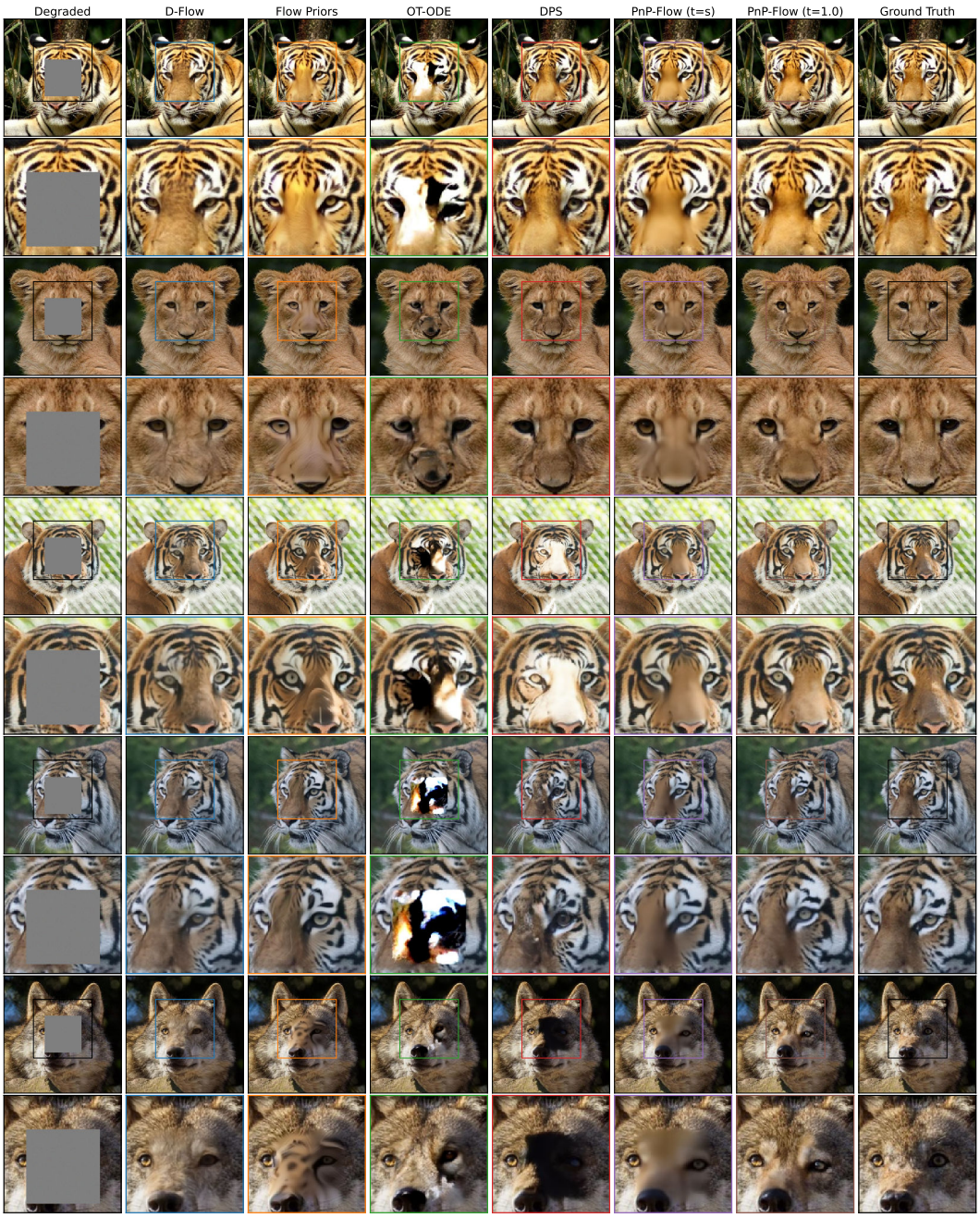


Figure 18: Box inpainting with mask 80×80, and $\sigma = 0.01$

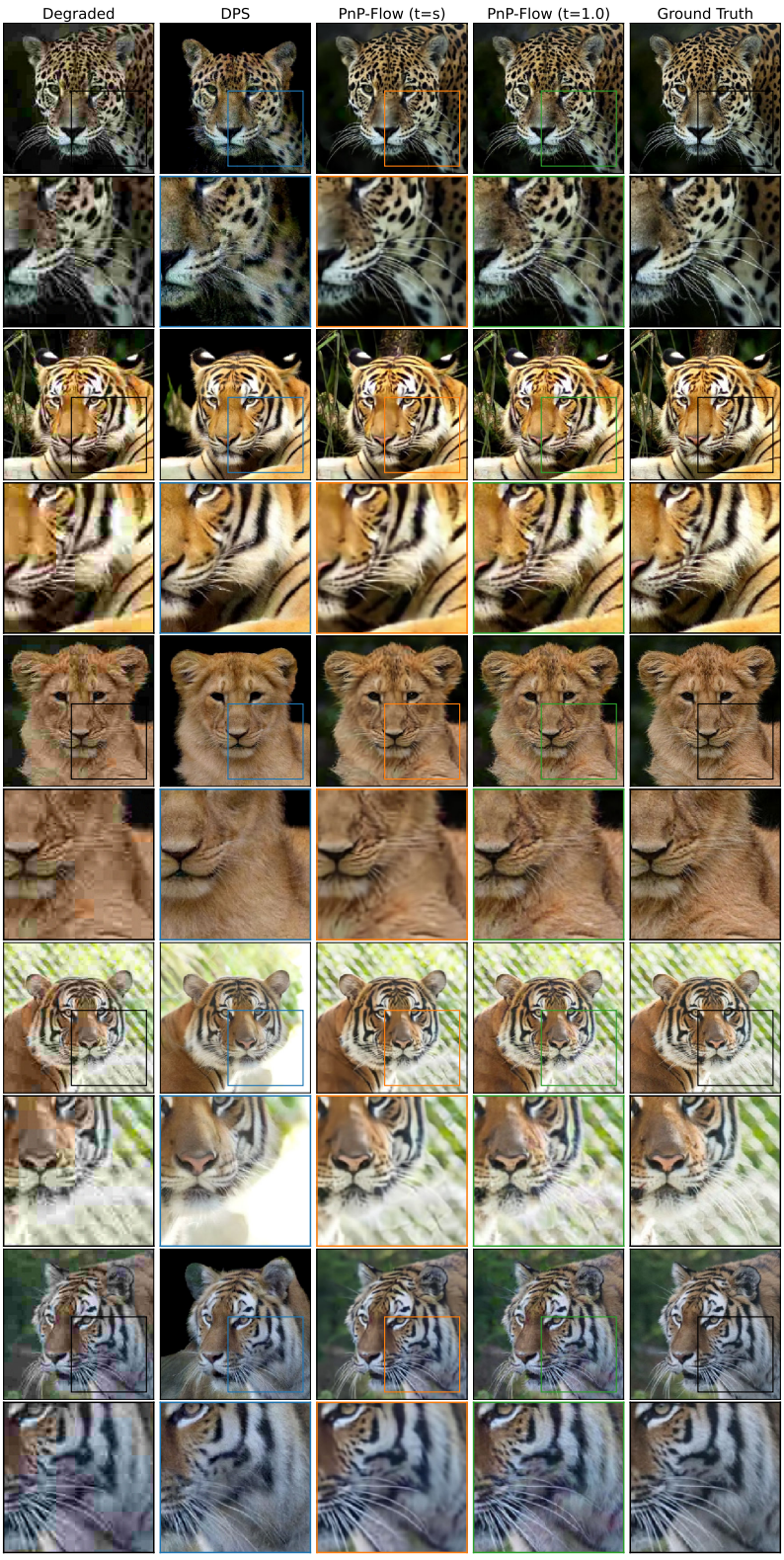


Figure 19: JPEG restoration.

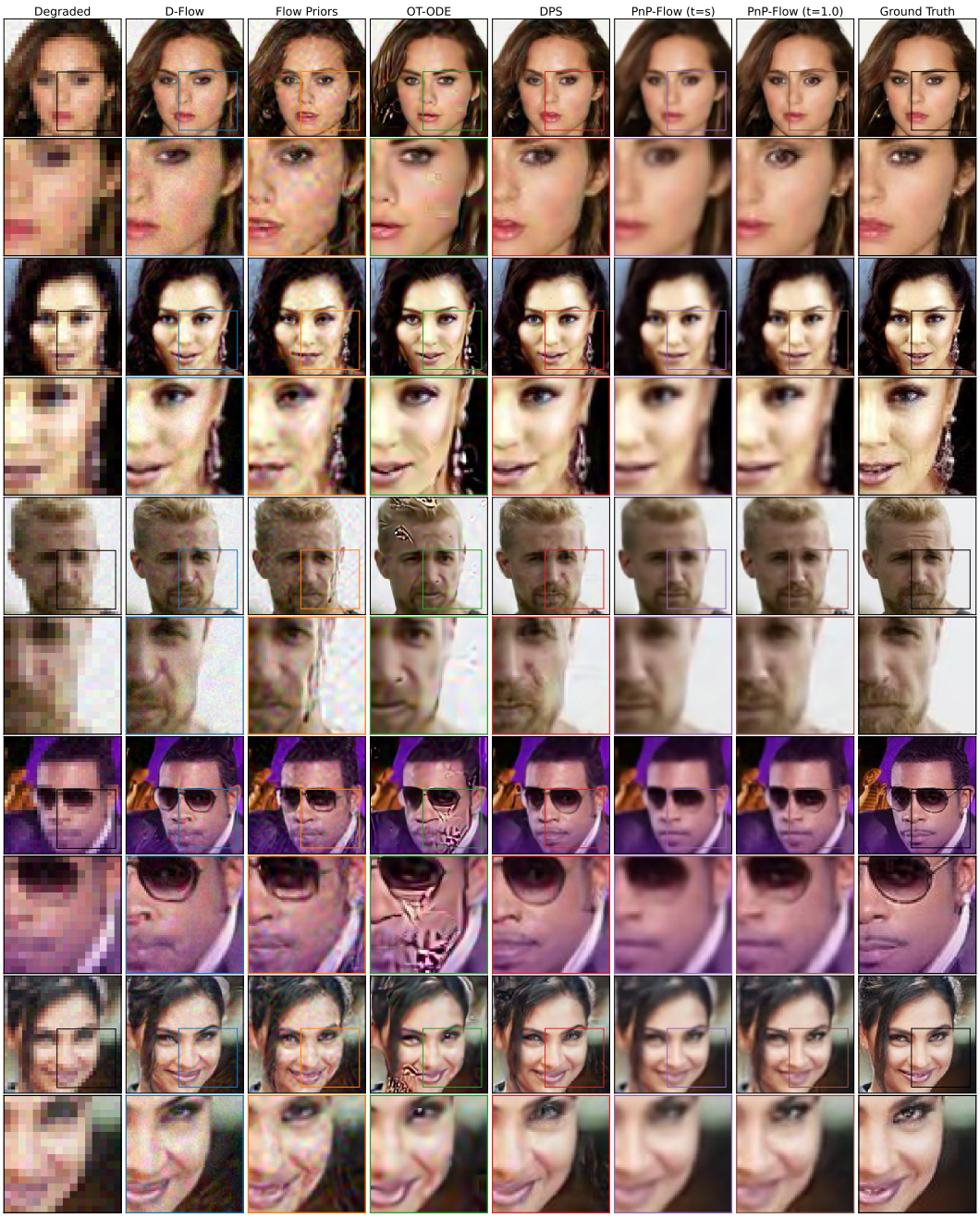


Figure 20: CelebA - Super resolution $\times 4$, and $\sigma = 0.05$

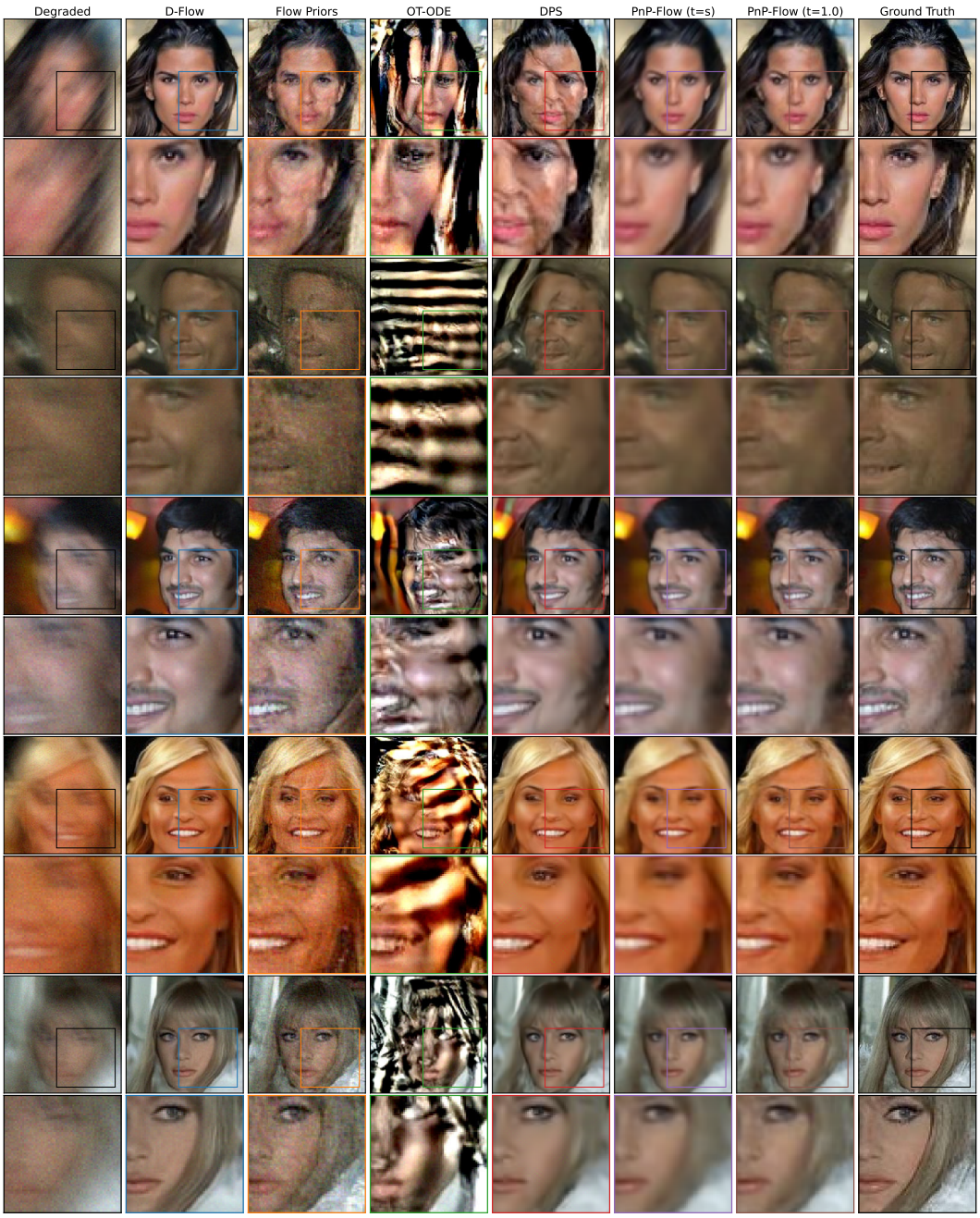


Figure 21: CelebA - Motion deblurring

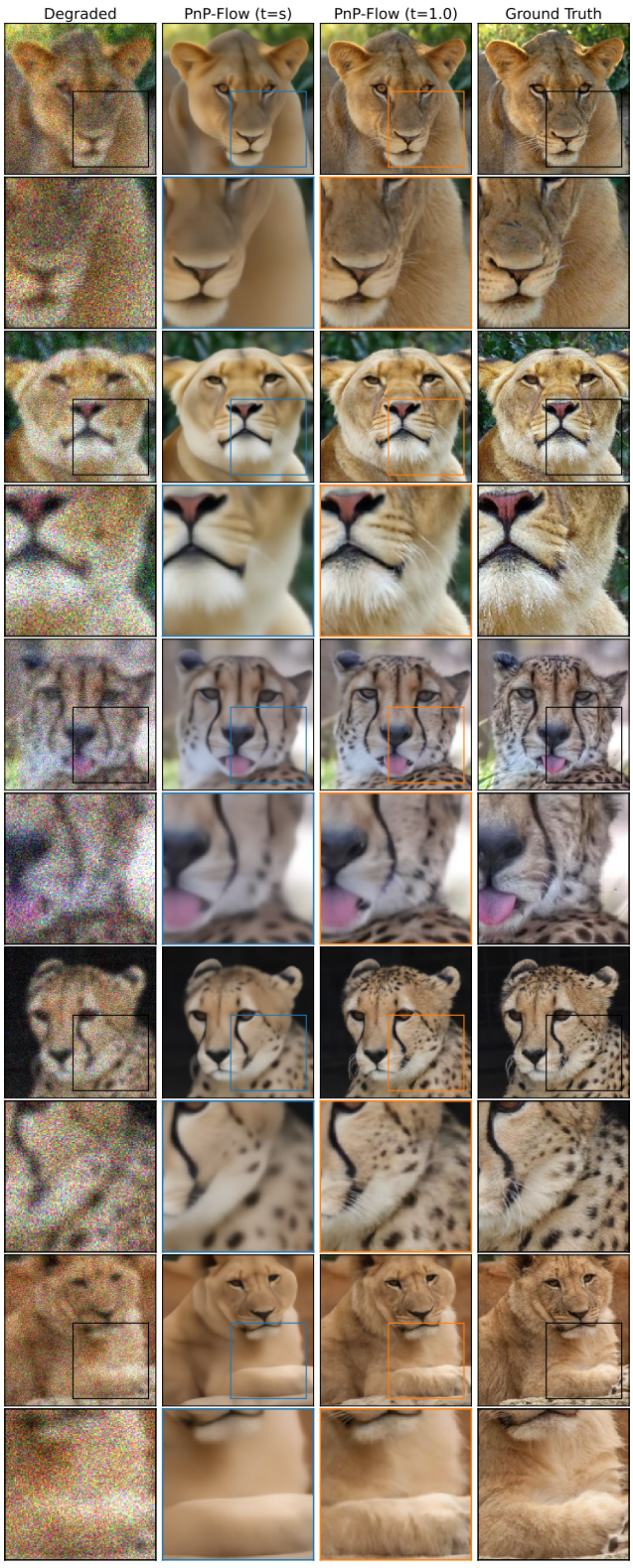


Figure 22: Gaussian deblurring with Poisson noise

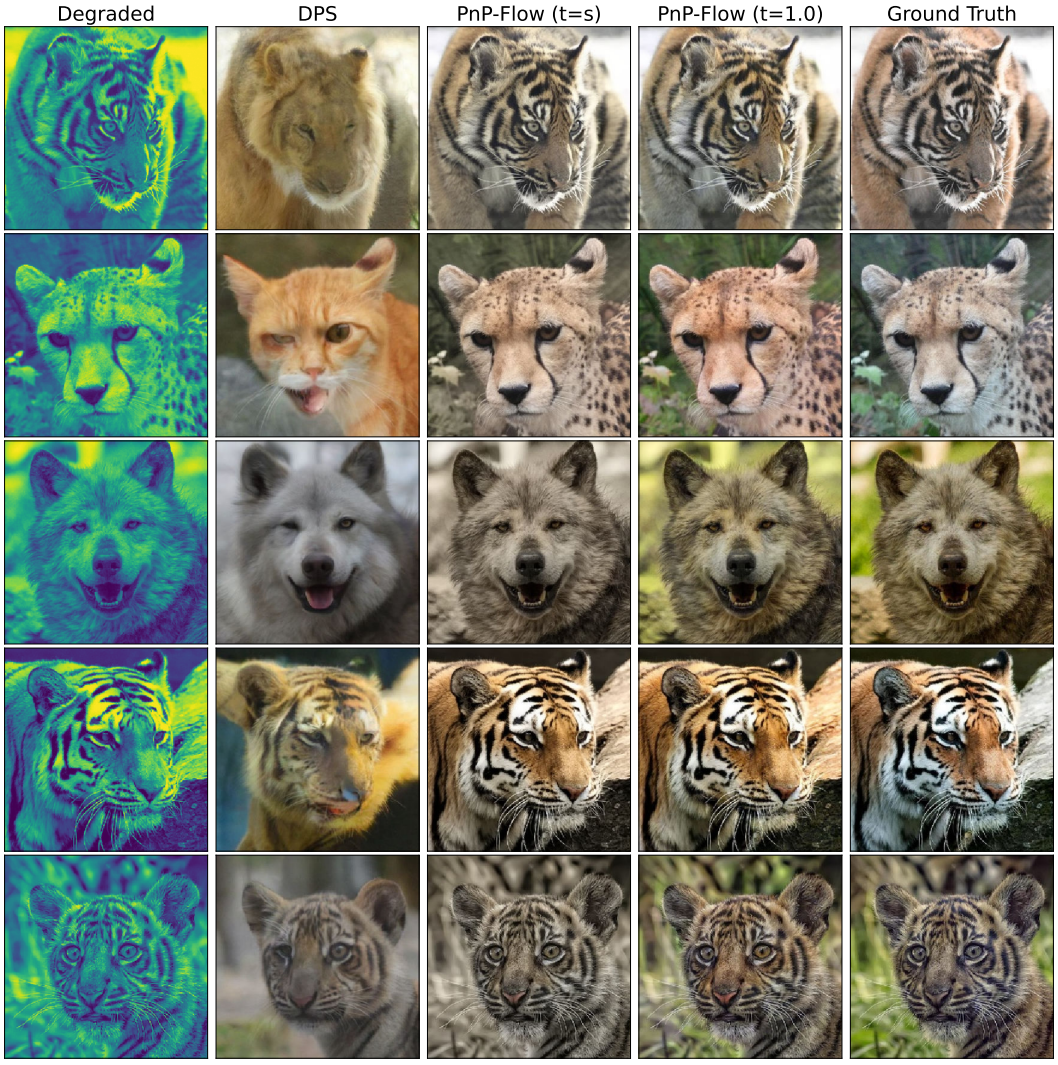


Figure 23: Colorization

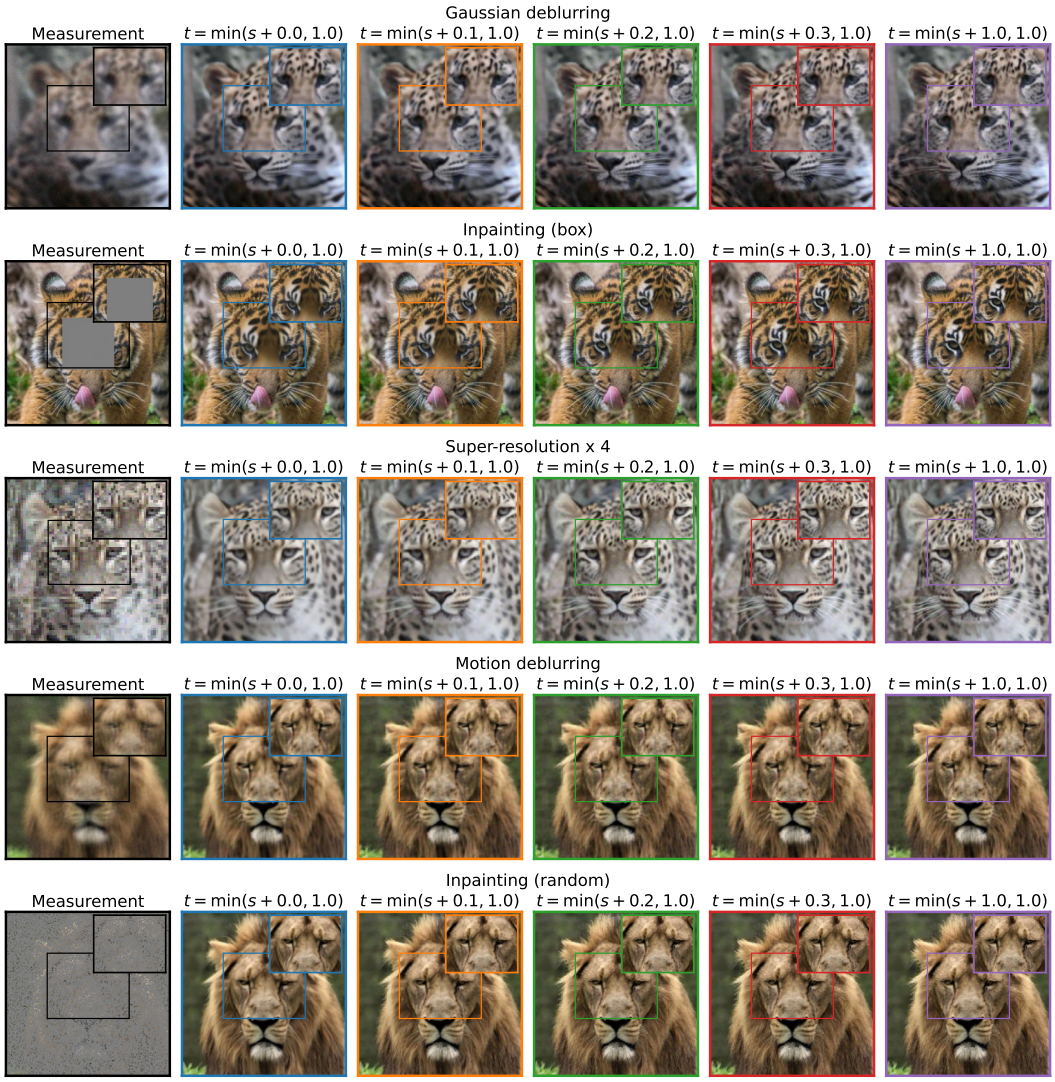


Figure 24: Larger version of the images in Fig. 3.

C.11 Visualization of the iterations for different lookaheads

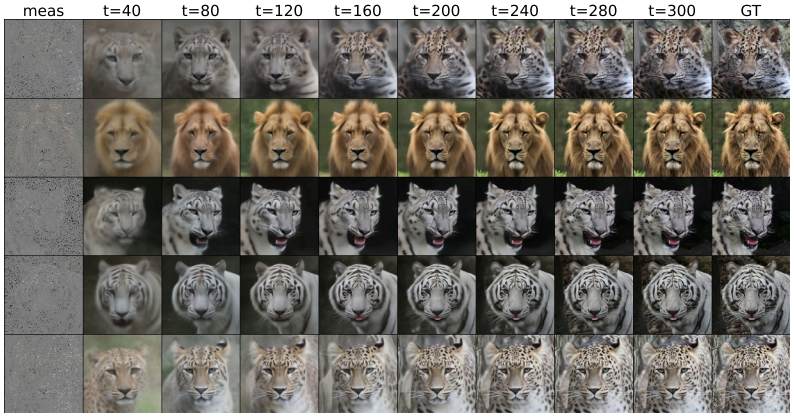


Figure 25: Progression of random inpainting with lookahead $t = s$

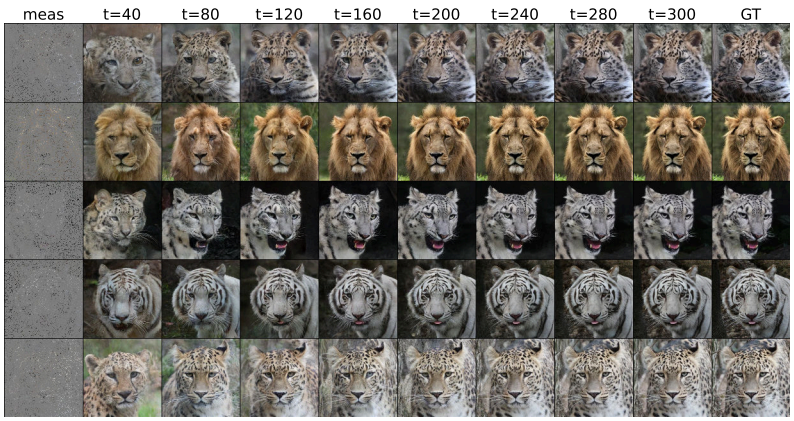


Figure 26: Progression of random inpainting with lookahead $t = 1$

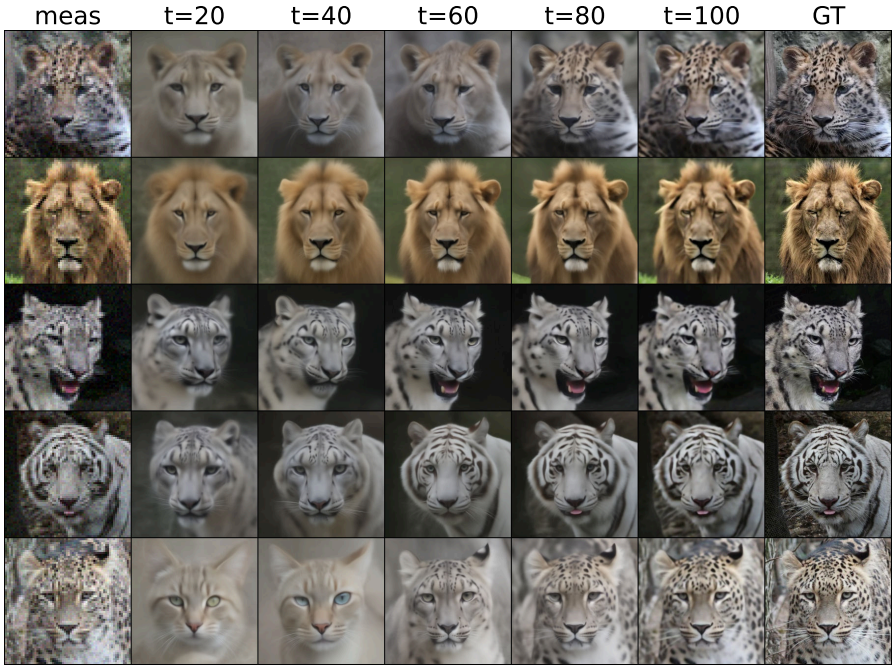


Figure 27: Progression of SR $\times 4$ with lookahead $t = s$

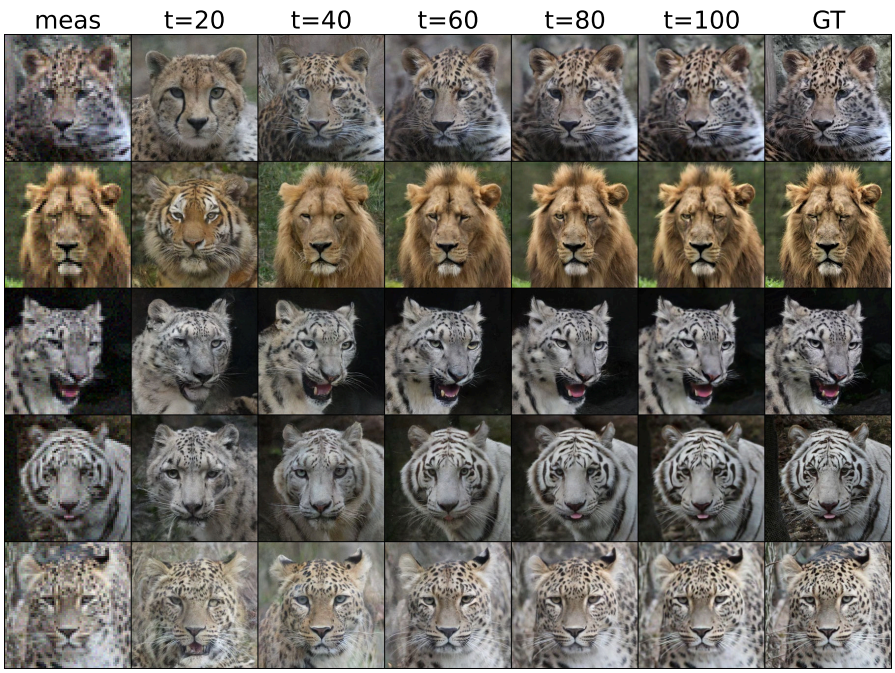


Figure 28: Progression of SR $\times 4$ with lookahead $t = 1$

C.12 Performance as a function of the number of PnP iterations

We study how the two endpoint lookaheads, $t = s$ and $t = 1$, depend on the number of PnP iterations N. Results for AFHQ are reported on super-resolution $\times 4$ (Fig. 29), Gaussian deblurring (Fig. 30) and box inpainting (Fig. 31); we also include for CelebA in Fig. 32.

The two regimes behave qualitatively differently. At $t = s$, both RMSE and FID decrease monotonically with N and plateau. At $t = 1$, the perceptual trajectory is non-monotonic: FID drops sharply within the first few tens of iterations, reaches its minimum, and then drifts back up as N continues to grow. The effect is most pronounced for Gaussian deblurring, where the best FID at $t = 1$ (≈ 11 at $N \approx 20$) is reached an order of magnitude earlier than the best FID at $t = s$ (≈ 23.5 at $N \approx 300$). This asymmetry is consistent with the measured growth of the local Lipschitz constant with t (Appendix D.4), which reflects the transition from an averaging MMSE estimator at $t = s$ to a less regular, perceptually oriented estimator at $t = 1$, so that approximation errors of the flow map compound more strongly over iterations at large lookaheads. RMSE remains well behaved in both cases, indicating that the drift is along the perceptual axis.

The practical takeaway is that the two endpoints call for different iteration schemes: the perceptual endpoint ($t = 1$) is best run for fewer steps than the distortion endpoint ($t = s$). The values of N used in our main experiments (Appendix B.2) reflect this asymmetry. This behavior is also consistent with the use of flow maps and consistency models as denoisers for few-step restoration (Spagnoletti et al., 2025; Gülle et al., 2026).

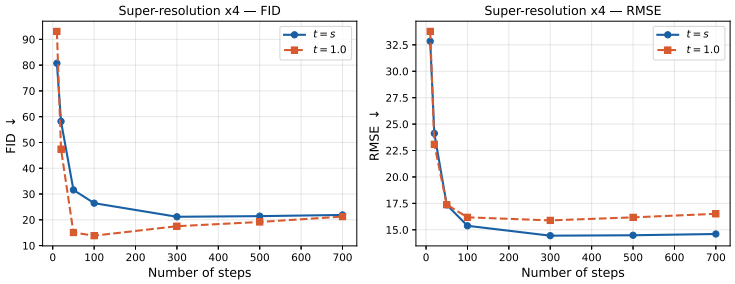


Figure 29: AFHQ - Super-resolution $\times 4$. Ablation of PnP-flow as a function of the number of steps.

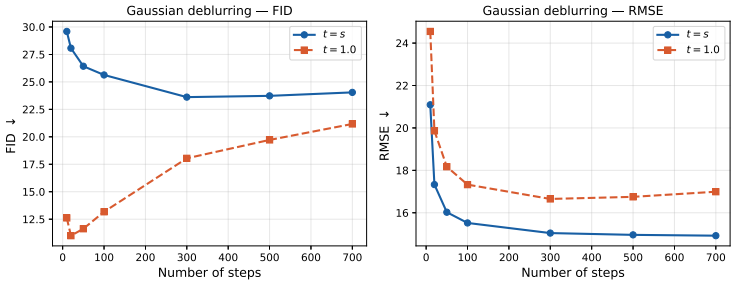


Figure 30: AFHQ - Gaussian deblurring. Ablation of PnP-flow as a function of the number of steps.

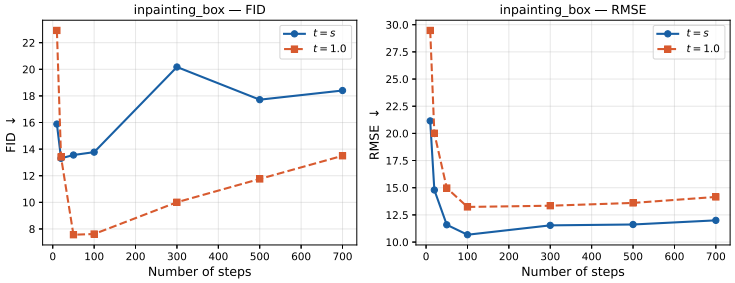


Figure 31: AFHQ - Box inpainting. Ablation of PnP-flow as a function of the number of steps.

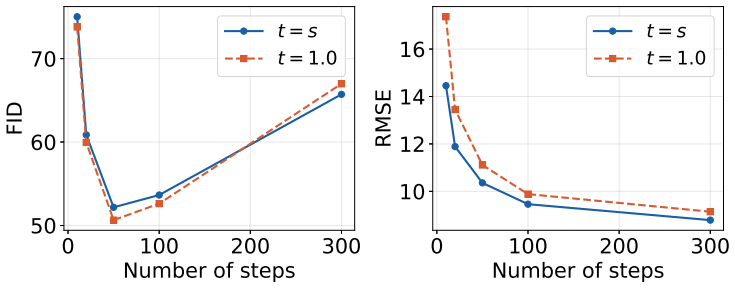


Figure 32: CelebA - Motion deblurring. Ablation of PnP-flow as a function of the number of steps.

C.13 Failure modes and limitations

For some degradations, the DP traversal is not monotone. We observe two such cases, box inpainting (Fig. 3) and phase retrieval, which point to two distinct mechanisms.

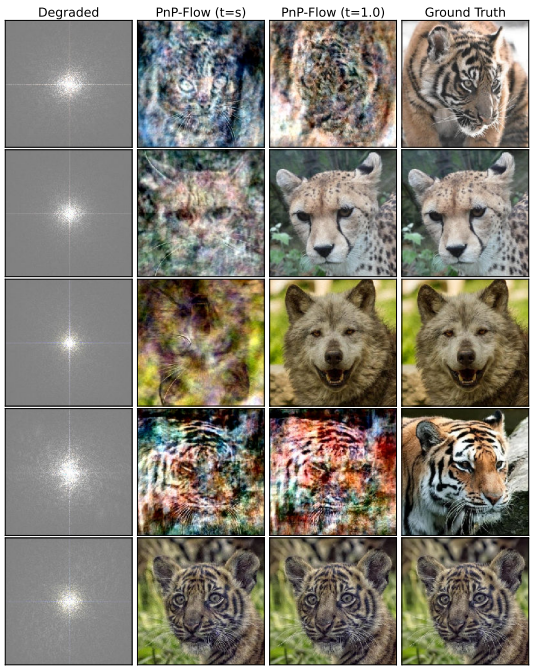


Figure 34: Reconstruction for phase retrieval

Phase retrieval. For phase retrieval with oversampling factor 4, PnP with the perceptual endpoint ($t = 1$) outperforms the MMSE endpoint ($t = s$) only after a certain number of iterations (Fig. 33), and the overall performance of both is limited (Table 14 and Fig. 34).

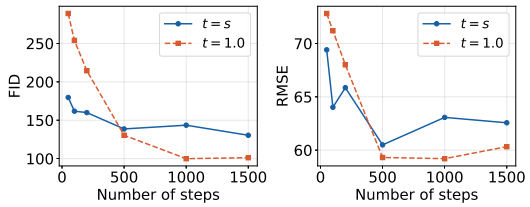


Figure 33: Performance as a function of the number of steps

Table 14: Quantitative results for Phase retrieval w/$\sigma = 0.01$ on AFHQ datasets. Best results are in **bold**.

Sampler	AFHQ 256×256			
	PSNR↑	LPIPS↓	FID↓	DISTS↓
PnP-Flow ($t = s$)	**17.36**	0.454	143.55	0.316
PnP-Flow ($t = 1$)	20.88	**0.317**	**99.95**	**0.240**

Mechanism 1: spatially coherent null spaces. The size of the null space of the forward operator alone does not explain the non-monotonicity: random inpainting with 90% missing pixels has a much larger null space than an 80×80 box, yet its DP curve is nearly monotone. We hypothesize that the *spatial coherence* of the null space is the dominant factor. For a random mask, the missing content is local and the prior interpolates it from dense surrounding context. A box mask instead forces the prior to synthesize coherent content over a contiguous region. As t increases, the denoiser shifts from posterior-mean-like toward sample-like outputs precisely in this region; the synthesized content is plausible under the prior, but the data-fidelity step cannot correct it, since the residual is blind to the null space. Flow-map approximation error, which grows with the lookahead (Appendix D.4), then determines how far this synthesis drifts from the observed context.

To test this hypothesis, we fix the fraction of missing pixels and vary the block size of the mask from 1×1 (random pixels) to 8×8 and 32×32 blocks, tracing the DP curve over the lookahead on AFHQ (Table 15). RMSE is monotone in t in all conditions, so any non-monotonicity is purely perceptual. For the random mask, the FID curve is monotone and then flat (reversal $+0.02$). As coherence increases, the curve becomes non-monotone, with a reversal of $+1.29$ for 8×8 blocks and $+2.34$ for 32×32 blocks; KID shows the same ordering (reversals of approximately 0, $0.9 \cdot 10^{-3}$, and $1.5 \cdot 10^{-3}$). Coherence also degrades the MMSE endpoint itself (FID 16.5 versus ≈ 77 at $t = s$), consistent with synthesis being intrinsically harder than interpolation. At fixed missingness, the monotonicity of the DP curve thus degrades continuously with mask coherence.

Table 15: FID (↓) along the lookahead sweep for inpainting on AFHQ with a fixed fraction of missing pixels and increasing block size of the mask. The last column reports the reversal, i.e., the FID at $t = 1$ minus the minimum FID along the sweep.

Mask	$t = s$	$t = 0.1$	$t = 0.3$	$t = 0.5$	$t = 0.7$	$t = 1$	Reversal
1×1 (random)	16.48	13.02	12.99	12.99	13.01	13.01	$+0.02$
8×8	76.71	42.59	29.46	29.32	29.99	30.61	$+1.29$
32×32	76.81	50.76	45.98	45.35	43.96	46.30	$+2.34$

Mechanism 2: non-convex data fidelity. In phase retrieval, the forward operator is nonlinear, so $g(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - f(\mathbf{x})\|^2$ is non-convex with spurious stationary points. Since the local Lipschitz constant of the iteration grows with t (Appendix D.4), large lookaheads are precisely where the iteration is most likely to move between basins; the failure stems from the interaction of the solver dynamics with the non-convex landscape.

Summary. Monotone traversal is expected when (i) the forward operator is linear, so that the data fidelity is convex, and (ii) the null space is spatially incoherent with image structure, so that the prior interpolates rather than synthesizes. Degradations that violate (ii), such as large contiguous masks, are prone to the first mechanism, and those that violate (i), such as phase retrieval and other nonlinear operators, to the second. This is consistent with our findings across the evaluated tasks.

D Proofs and theoretical results

D.1 Average denoiser.

Recall the definition of the average denoiser

$$D_{s,t}(\mathbf{x}) = \mathbf{x} + (1 - s)\,\mathbf{v}(\mathbf{x}, s, t), \tag{15}$$

where $\mathbf{v}(\mathbf{x}, s, t)$ denotes the average velocity of the flow map over $[s, t]$, and the flow map is parameterized as

$$X_{s,t}(\mathbf{x}) = \mathbf{x} + (t - s)\,\mathbf{v}(\mathbf{x}, s, t). \tag{16}$$

Proposition 1 (Basic properties) *For all $0 \le s \le u \le t \le 1$, the average denoiser satisfies:*

1. ***Flow-map relation.***

$$X_{s,t}(\mathbf{x}) = \frac{1 - t}{1 - s}\,\mathbf{x} + \frac{t - s}{1 - s}\,D_{s,t}(\mathbf{x}). \tag{17}$$

2. ***Diagonal consistency (MMSE).***

$$D_{s,s}(\mathbf{x}) = D_s(\mathbf{x}) = \mathbb{E}[\mathbf{x}_1 \mid \mathbf{x}_s = \mathbf{x}]. \tag{18}$$

3. ***Semigroup structure.***

$$D_{s,t}(\mathbf{x}) = \gamma\,D_{s,u}(\mathbf{x}) + (1 - \gamma)\,D_{u,t}(X_{s,u}(\mathbf{x})), \tag{19}$$

with

$$\gamma = \frac{(1 - t)(u - s)}{(1 - u)(t - s)} \in [0, 1]. \tag{20}$$

Proof. **(1) Flow-map relation.** Using the parameterization,

$$D_{s,t}(\mathbf{x}) - \mathbf{x} = (1-s)\,\mathbf{v}(\mathbf{x}, s, t), \quad X_{s,t}(\mathbf{x}) - \mathbf{x} = (t-s)\,\mathbf{v}(\mathbf{x}, s, t).$$

Eliminating $\mathbf{v}$ gives

$$X_{s,t}(\mathbf{x}) = \mathbf{x} + \frac{t-s}{1-s}\big(D_{s,t}(\mathbf{x}) - \mathbf{x}\big) = \frac{1-t}{1-s}\,\mathbf{x} + \frac{t-s}{1-s}\,D_{s,t}(\mathbf{x}).$$

(2) Diagonal consistency. By definition of the flow map,

$$\lim_{t \to s} \mathbf{v}(\mathbf{x}, s, t) = \mathbf{v}(\mathbf{x}, s),$$

the instantaneous velocity. Using Tweedie's relation,

$$D_s(\mathbf{x}) = \mathbf{x} + (1-s)\,\mathbf{v}(\mathbf{x}, s),$$

which matches $D_{s,t}$ at $t = s$.

(3) Semigroup structure. From the flow map semigroup property,

$$X_{s,t}(\mathbf{x}) = X_{u,t}(\mathbf{X}_{s,u}(\mathbf{x})).$$

Substitute the affine forms from (i) for each map and match coefficients of $\mathbf{x}$ and $D_{s,t}$. After rearrangement, this yields the convex combination:

$$D_{s,t}(\mathbf{x}) = \gamma\, D_{s,u}(\mathbf{x}) + (1-\gamma)\, D_{u,t}(X_{s,u}(\mathbf{x})),$$

with γ as stated.

D.2 Full derivation for the Gaussian case (Theorem 1)

We provide the complete derivation of the average denoiser gain and its connection to the optimal DP estimator for the scalar Gaussian target.

D.2.1 Setup

Let $p_1 = \mathcal{N}(0, \sigma_p^2)$ with $\sigma_p^2 \le 1$, and let $x_\tau = \tau x_1 + (1-\tau)\epsilon$ with $x_1 \sim p_1$, $\epsilon \sim \mathcal{N}(0,1)$, and $x_1 \perp \epsilon$. Then $x_\tau \sim \mathcal{N}(0, \sigma_\tau^2)$ with $\sigma_\tau^2 := \tau^2\sigma_p^2 + (1-\tau)^2$. The instantaneous denoiser and velocity are linear:

$$D_\tau(x) = a_\tau\, x, \quad a_\tau = \frac{\tau\sigma_p^2}{\sigma_\tau^2}, \qquad v_\tau(x) = b_\tau\, x, \quad b_\tau = \frac{a_\tau - 1}{1-\tau} = \frac{\tau(\sigma_p^2+1) - 1}{\sigma_\tau^2}. \tag{21}$$

By linearity, the ODE $\dot{x}_\tau = v_\tau(x_\tau) = b_\tau x_\tau$ admits the closed-form solution $x_t = \Phi(t,s)\,x_s$ with $\Phi(t,s) = \exp\big(\int_s^t b_\tau\, d\tau\big)$. The average velocity over $[s,t]$ then satisfies $x_t = x_s + (t-s)\,v_{s,t}(x_s)$ with $v_{s,t}(x_s) = [\Phi(t,s) - 1]/(t-s) \cdot x_s$, and the average denoiser $D_{s,t}(x) = x + (1-s)\,v_{s,t}(x)$ becomes $D_{s,t}(x_s) = \Lambda(s,t)\,x_s$ with:

$$\Lambda(s,t) = \frac{(1-s)\,\Phi(t,s) - (1-t)}{t-s}. \tag{22}$$

We now introduce a few useful Lemmas.

Lemma 1 (Closed form for $\Phi(t,s)$ and $\Lambda(s,t)$) $\Phi(t,s) = \sigma_t/\sigma_s$.

Proof: A direct computation gives $\frac{d}{d\tau}\frac{1}{2}\log\sigma_\tau^2 = \dot{\sigma}_\tau^2/(2\sigma_\tau^2) = [\tau(\sigma_p^2+1) - 1]/\sigma_\tau^2 = b_\tau$. Hence $\int_s^t b_\tau\, d\tau = \log(\sigma_t/\sigma_s)$ and $\Phi(t,s) = \sigma_t/\sigma_s$. □

Substituting Lemma 1 into (22) yields the closed form stated in (9):

$$\Lambda(s,t) = \frac{(1-s)\sigma_t - (1-t)\sigma_s}{\sigma_s(t-s)}. \tag{23}$$

Boundary values. At $t = 1$, $\sigma_1 = \sigma_p$ gives directly $\Lambda(s, 1) = \sigma_p/\sigma_s$. At $t = s$, applying L'Hôpital to (9) and using $\dot{\sigma}_\tau = [\tau(\sigma_p^2 + 1) - 1]/\sigma_\tau$:

$$\Lambda(s, s) = \frac{(1 - s)\dot{\sigma}_s + \sigma_s}{\sigma_s} = 1 + \frac{(1 - s)[s(\sigma_p^2 + 1) - 1]}{\sigma_s^2} = \frac{s\sigma_p^2}{\sigma_s^2} = a_s, \tag{24}$$

where the last equality uses $\sigma_s^2 + (1 - s)[s(\sigma_p^2 + 1) - 1] = s\sigma_p^2$, which one verifies by expanding $\sigma_s^2 = s^2\sigma_p^2 + (1 - s)^2$.

Lemma 2 (Monotonicity of $\Lambda(s, t)$) *For fixed $s \in (0, 1)$, $t \mapsto \Lambda(s, t)$ is strictly increasing on $[s, 1]$.*

Proof: Differentiating (9) in t and writing $\sigma_t - \sigma_s = \int_s^t \dot{\sigma}_u \, du$:

$$\partial_t \Lambda(s, t) = \frac{1 - s}{\sigma_s (t - s)^2} \left[(t - s)\dot{\sigma}_t - \int_s^t \dot{\sigma}_u \, du \right]. \tag{25}$$

Writing $\sigma_\tau^2 = c\tau^2 - 2\tau + 1$ with $c = \sigma_p^2 + 1$, a direct calculation gives $\ddot{\sigma}_\tau = \left[c\,\sigma_\tau^2 - (c\tau - 1)^2\right]/\sigma_\tau^3 = \sigma_p^2/\sigma_\tau^3 > 0$ for all $\tau \in (0, 1)$, so $\dot{\sigma}_\tau$ is strictly increasing. Therefore $\int_s^t \dot{\sigma}_u \, du < (t - s)\dot{\sigma}_t$ for $s < t$, giving $\partial_t \Lambda > 0$. □

On the assumption $\sigma_p^2 \leq 1$. The condition $\sigma_p^2 \leq 1$ is a normalization convention, stating that the target variance does not exceed that of the standard normal source, as for data scaled to $[-1, 1]$. The arguments above do not use it: the convexity $\ddot{\sigma}_\tau = \sigma_p^2/\sigma_\tau^3 > 0$ holds for any $\sigma_p > 0$, and $\sigma_p/\sigma_s > a_s$ is equivalent to $\sigma_s^2 > s^2\sigma_p^2$, which holds for every $s < 1$. Hence Lemma 2, Proposition 2, and Theorem 1 remain valid for any $\sigma_p > 0$.

D.2.2 Connection to the optimal DP estimator

The DP optimal estimator from Freirich et al. (2021). The MMSE estimator $\hat{X}_0 = a_s x_s$ has standard deviation $\hat{\sigma} = a_s\sigma_s = s\sigma_p^2/\sigma_s$, and its perceptual gap to the prior is $P^* = W_2(p_{\hat{X}_0}, p_1) = |\sigma_p - \hat{\sigma}|$. The W_2-optimal transport map from $\mathcal{N}(0, \hat{\sigma}^2)$ to $p_1 = \mathcal{N}(0, \sigma_p^2)$ is the scaling $T^* = \sigma_p/\hat{\sigma}$. By Freirich et al. (2021, Theorem 3), the minimum-distortion deterministic estimator at perception level $P = \alpha P^*$ ($\alpha \in [0, 1]$) is the displacement interpolant

$$\hat{X}_\alpha^{\text{det}} = \left[\alpha + (1 - \alpha)T^*\right] a_s \, x_s =: \Gamma_s(\alpha) \, x_s, \tag{26}$$

with $\Gamma_s(1) = a_s$ (MMSE) and $\Gamma_s(0) = T^* a_s = \sigma_p/\sigma_s$ (perfect perception). The corresponding distortion-perception function is $D(P) = D^* + (P^* - P)^2$.

Proposition 2 (Average denoiser is exactly optimal) *For any $s \in (0, 1)$ and $t \in [s, 1]$, $\Lambda(s, t) = \Gamma_s(\alpha(s, t))$ where*

$$\alpha(s, t) = \frac{\sigma_p/\sigma_s - \Lambda(s, t)}{\sigma_p/\sigma_s - a_s} \tag{27}$$

decreases monotonically from 1 at $t = s$ to 0 at $t = 1$.

Proof: By (26), Γ is affine in α and maps $[0, 1]$ bijectively onto $[a_s, \sigma_p/\sigma_s]$ (since $\sigma_p/\sigma_s > a_s$ whenever $s < 1$). By the boundary values of Λ and Lemma 2, $\Lambda(s, \cdot)$ is continuous and strictly increasing on $[s, 1]$ with $\Lambda(s, s) = a_s = \Gamma_s(1)$ and $\Lambda(s, 1) = \sigma_p/\sigma_s = \Gamma_s(0)$. Inverting the affine relation $\Gamma_s(\alpha) = \Lambda(s, t)$ yields $\alpha(s, t)$ as stated, and the monotonicity follows because Γ is decreasing in α while Λ is increasing in t. □

Summary. As t increases from s to 1, the average denoiser gain $\Lambda(s, t)$ traces the affine segment from the MMSE gain a_s to the perfect-perception gain σ_p/σ_s. This segment parametrizes the W_2 geodesic from the MMSE output distribution $p_{\hat{X}_0}$ to the target p_1, and by Freirich et al. (2021, Theorem 3), every point on it achieves the minimum distortion at its perception level. A visual comparison on images is shown in Fig. 35.

D.2.3 Multivariate Gaussian targets

Isotropic and correlated targets. Let $p_1 = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}_\Sigma\mathbf{U}^\top$ and $\mathbf{\Lambda}_\Sigma = \mathrm{diag}(\sigma_{p,1}^2, \ldots, \sigma_{p,d}^2)$. Since the source is isotropic, $\mathbf{U}^\top\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, so the rotated variable $\tilde{\mathbf{x}}_\tau = \mathbf{U}^\top\mathbf{x}_\tau$ is the linear interpolant between $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_\Sigma)$, whose coordinates are independent. The velocity, flow map, and average denoiser are equivariant under this rotation, so the problem reduces to d independent scalar problems, each described by Appendix D.2 with its own $\sigma_{p,i}$, gains $a_{s,i}$ and $\Lambda_i(s,t)$, and perception level

$$\alpha_i(s,t) = \frac{\sigma_{p,i}/\sigma_{s,i} - \Lambda_i(s,t)}{\sigma_{p,i}/\sigma_{s,i} - a_{s,i}}. \tag{28}$$

In particular, for isotropic targets ($\sigma_{p,i} \equiv \sigma_p$), all coordinates share the same $\alpha(s,t)$ and Theorem 1 holds verbatim.

Anisotropic targets. The optimal DP estimator at perception level P (Freirich et al., 2021, Theorem 3) applies the *same* interpolation weight $\alpha = P/P^*$ in every direction, whereas a shared lookahead induces the direction-dependent weights $\alpha_i(s,t)$. Consequently:

- the endpoints are exactly optimal for any covariance: $t = s$ gives $\alpha_i \equiv 1$ (MMSE) and $t = 1$ gives $\alpha_i \equiv 0$ (perfect perception);
- intermediate lookaheads are exactly optimal if and only if all $\alpha_i(s,t)$ coincide, which holds when the spectrum is flat;
- otherwise, the deviation from optimality is governed by the spread $\max_i \alpha_i(s,t) - \min_i \alpha_i(s,t)$, which grows with the spread of the spectrum and vanishes as it flattens.

A single scalar lookahead, as used in Algorithm 1, therefore does not realize the optimal DP curve for anisotropic targets at intermediate operating points.

Recovering the optimal curve by endpoint interpolation. Exact optimality is nevertheless attainable from the same trained model. The endpoint $D_{s,s}$ is the MMSE estimator, and $D_{s,1}$ coincides with the optimal perfect-perception estimator (Freirich et al., 2021, Theorem 2): the W_2-optimal map from the distribution of the MMSE estimate to p_1 scales direction i by $\sigma_{p,i}/(a_{s,i}\sigma_{s,i})$, which applied to $a_{s,i}\tilde{x}_{s,i}$ gives exactly $(\sigma_{p,i}/\sigma_{s,i})\,\tilde{x}_{s,i} = \Lambda_i(s,1)\,\tilde{x}_{s,i}$. Since the optimal DP estimators form the convex combination of these two endpoints (Freirich et al., 2021, Theorem 3), the family

$$\hat{\mathbf{x}}_\lambda = (1-\lambda)\,D_{s,1}(\mathbf{x}_s) + \lambda\,D_{s,s}(\mathbf{x}_s), \qquad \lambda \in [0,1], \tag{29}$$

traces the optimal DP curve exactly for any Gaussian target, with $\alpha_i = \lambda$ in every direction, at the cost of two network evaluations instead of one. For isotropic targets, $\Lambda(s,t)$ is itself a convex combination of the endpoint gains, so the lookahead and (29) trace the same curve under different parametrizations. Appendix C.6 compares both constructions on images.

Empirical (discrete) target distributions. For an empirical target $p_1 = \frac{1}{n}\sum_i \delta_{\mathbf{x}^{(i)}}$, the instantaneous denoiser is a softmax-weighted average of the atoms $\mathbf{x}^{(i)}$. At $t = s$, $D_{s,s}$ returns this average (the empirical analogue of the blurry MMSE estimate), while at $t = 1$ it returns the endpoint of the ODE, a single atom, so that the perception index is zero. The qualitative DP behavior therefore persists (see also Fig. 1). However, we cannot claim optimality at intermediate lookaheads: the proof of Theorem 1 relies on the linearity of $D_{s,t}$, which fails here, and the path induced by the ODE need not follow the W_2 geodesic.



Figure 35: Visual comparison for $\xi = 0.3$. From left to right: clean image, noisy input, and reconstructions at increasing lookahead. The estimator smoothly transitions from a blurry conditional mean (MMSE) to a sharp sample, progressively restoring texture, contrast, and color saturation.

D.3 Experiment on Mixture-of-Gaussians

D.3.1 MMSE denoiser for a mixture of Gaussians

Setup. Let $p_0 = \mathcal{N}(0, \mathbf{I})$ and $p_1 = \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_a, \sigma_c^2\mathbf{I}) + \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_b, \sigma_c^2\mathbf{I})$. Under the linear interpolant $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$, we seek to compute the MMSE denoiser:

$$D_t(\mathbf{x}) = \mathbb{E}[\mathbf{x}_1 \mid \mathbf{x}_t = \mathbf{x}]. \tag{30}$$

Step 1: Marginal distribution of $\mathbf{x}_t$. Conditioned on $\mathbf{x}_1$ originating from component $k \in \{a, b\}$, we have $\mathbf{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_k, \sigma_c^2\mathbf{I})$ and $\mathbf{x}_0 \sim \mathcal{N}(0, \mathbf{I})$. Thus:

$$\mathbf{x}_t \mid \text{component } k \;\sim\; \mathcal{N}\big(t\,\boldsymbol{\mu}_k,\; \sigma_t^2\mathbf{I}\big)\,, \quad \text{where } \sigma_t^2 = (1-t)^2 + t^2\sigma_c^2. \tag{31}$$

The marginal distribution is the mixture:

$$p(\mathbf{x}_t) = \frac{1}{2}\,\mathcal{N}(\mathbf{x}_t;\, t\boldsymbol{\mu}_a,\, \sigma_t^2\mathbf{I}) + \frac{1}{2}\,\mathcal{N}(\mathbf{x}_t;\, t\boldsymbol{\mu}_b,\, \sigma_t^2\mathbf{I}). \tag{32}$$

Step 2: Posterior weights. By Bayes' rule, the posterior probability $w_k(\mathbf{x}, t)$ that $\mathbf{x}_1$ originated from component k given $\mathbf{x}_t = \mathbf{x}$ is:

$$w_k(\mathbf{x}, t) = \frac{\mathcal{N}(\mathbf{x};\, t\boldsymbol{\mu}_k,\, \sigma_t^2\mathbf{I})}{\mathcal{N}(\mathbf{x};\, t\boldsymbol{\mu}_a,\, \sigma_t^2\mathbf{I}) + \mathcal{N}(\mathbf{x};\, t\boldsymbol{\mu}_b,\, \sigma_t^2\mathbf{I})}. \tag{33}$$

In log-space, this is proportional to $-\frac{\|\mathbf{x}-t\boldsymbol{\mu}_k\|^2}{2\sigma_t^2}$, and the weights are normalized via softmax such that $w_a + w_b = 1$.

Step 3: Conditional mean within components. Given that $\mathbf{x}_1$ comes from component k, the posterior mean of $\mathbf{x}_1$ given $\mathbf{x}_t = \mathbf{x}$ is:

$$\mathbb{E}[\mathbf{x}_1 \mid \mathbf{x}_t = \mathbf{x}, \text{component } k] = \boldsymbol{\mu}_k + \kappa_t(\mathbf{x} - t\,\boldsymbol{\mu}_k), \tag{34}$$

where $\kappa_t = \frac{t\,\sigma_c^2}{\sigma_t^2}$ is the regression coefficient (Tweedie's formula applied component-wise).

Step 4: Full MMSE denoiser. By the law of total expectation, the denoiser is the weighted average:

$$D_t(\mathbf{x}) = w_a(\mathbf{x}, t)\ [\boldsymbol{\mu}_a + \kappa_t(\mathbf{x} - t\boldsymbol{\mu}_a)] + w_b(\mathbf{x}, t)\ [\boldsymbol{\mu}_b + \kappa_t(\mathbf{x} - t\boldsymbol{\mu}_b)]\,. \tag{35}$$

This is a softmax-weighted average of the within-component posterior means. Its behavior is intuitive:

- When $\mathbf{x}$ is close to $t\boldsymbol{\mu}_a$, $w_a \approx 1$ and $D_t(\mathbf{x}) \approx \boldsymbol{\mu}_a + \kappa_t(\mathbf{x} - t\boldsymbol{\mu}_a)$ (assigned to component a).
- When $\mathbf{x}$ is equidistant from $t\boldsymbol{\mu}_a$ and $t\boldsymbol{\mu}_b$, $w_a \approx w_b \approx 0.5$, and $D_t(\mathbf{x})$ averages the two, producing an output between the modes (the posterior averaging artifact).

We compute the average denoiser by ODE integration using the instantaneous velocity field. The nonlinearity introduced by the softmax weights makes the MoG fundamentally different from the single Gaussian case: the denoiser is non-linear in $\mathbf{x}$, the flow map lacks a closed-form solution, and the posterior averaging artifact (blurring between modes) emerges as a direct consequence of the mixture structure. However, given the simplicity of the instantaneous denoiser, we can approximate the average denoiser numerically.

D.3.2 DP analysis

In this section, we consider a controlled setting based on a 2D mixture of Gaussians. For this distribution, the marginal velocity admits a closed-form expression, allowing both the average denoiser $D_{s,t}$ and the optimal DP estimator of Freirich et al. (2021) to be evaluated with high precision, bypassing the requirement of a learned model.

Setup. We consider $\boldsymbol{\mu}_a = (-2, 0)$, $\boldsymbol{\mu}_b = (2, 0)$, and $\sigma_c = 0.3$, and compute the average denoiser $D_{s,t}$ by integrating $\dot{x}_\tau = v(x_\tau, \tau)$ from $\tau = s$ to $\tau = t$ with a fourth-order Runge–Kutta scheme (200 steps).

Optimal DP construction. To construct the optimal DP curve, we follow Freirich et al. (2021, Theorem 3). At perception level $P \in [0, P^*]$, the optimal estimator is given by (7).

We approximate this construction empirically by drawing $N = 5000$ samples, computing $\hat{\mathbf{x}}_0^{(i)} = D_s(\mathbf{x}_s^{(i)})$, and solving the discrete optimal transport problem between the empirical distributions $\{\hat{\mathbf{x}}_0^{(i)}\}$ and $\{\mathbf{x}_1^{(j)}\} \sim p_1$ using (Flamary et al., 2021). Let π denote the optimal coupling; we then form the interpolated samples as

$$\hat{\mathbf{x}}_P^{(i)} = \left(1 - \frac{P}{P^*}\right) \hat{\mathbf{x}}_0^{(i)} + \frac{P}{P^*} \mathbf{x}_1^{(\pi(i))}.$$

Distortion is measured as MSE with respect to the ground truth, while perception is quantified via the squared 2-Wasserstein distance $W_2^2(\hat{p}, p_1)$ between empirical distributions.

Results. Fig. 36 compares the resulting curves at three noise levels $s \in 0.3, 0.5, 0.7$, corresponding to heavy, moderate, and light noise (recall that smaller s implies heavier noise, with $\xi := 1 - s$). Three observations stand out.

1. **Endpoints coincide at MMSE.** At $t = s$ (equivalently, $\alpha = 1$), both curves reduce to the MMSE estimator by construction, yielding identical distortion–perception coordinates across all panels.
2. **Near-optimal behavior in the interior.** For intermediate lookaheads, the average denoiser traces a DP curve that lies slightly above the Freirich optimum, with a small and nearly uniform gap. At $s = 0.3$ (heavy noise), this gap remains below 0.02 in MSE across the full perception range. This discrepancy is not a numerical artifact: it persists when using higher-order integration (RK4) and when increasing the sample size from $N = 2000$ to $N = 5000$. Instead, it reflects a structural limitation—while the flow-map trajectory matches the W_2 geodesic in the Gaussian case, it deviates slightly for nonlinear velocity fields (here induced by the softmax-weighted mixture).

3. **Endpoint behavior at $t = 1$.** At the perceptual endpoint, the distribution $D_{s,1}(p_{x_s})$ closely approximates p_1 but does not match it exactly, resulting in a small residual W_2^2 gap. This gap decreases as noise increases (from 9.4×10^{-3} at $s = 0.5$ to 3.8×10^{-3} at $s = 0.3$, compared to a sampling floor of 2.2×10^{-3}). This is consistent with the intuition that the integrable $1/(1-\tau)$ singularity in the velocity field has a stronger effect when the final integration interval is short.

Taken together, these results show that the average denoiser is *near-optimal* in a controlled setting: the resulting curves are smooth, monotone, and convex, match the MMSE endpoint exactly, and exhibit only small, well-characterized deviations from the Freirich optimum. This complements the exact optimality established in the Gaussian case (Theorem 1) and supports the behavior observed at image scale (Fig. 2a).

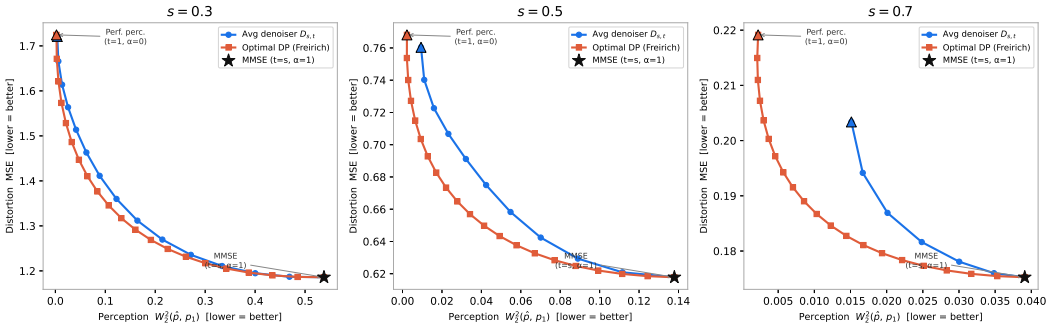


Figure 36: Distortion–perception comparison on a 2D mixture of Gaussians at three noise levels $s \in \{0.3, 0.5, 0.7\}$ (heavy, moderate, light noise). Blue: average denoiser $D_{s,t}$ swept over lookahead $t \in [s, 1]$. Red: optimal DP estimator of Freirich et al. (2021) swept over perception level $\alpha \in [0, 1]$, computed via exact W_2-optimal transport between the MMSE pushforward and p_1. Both curves share the MMSE endpoint exactly (★); small residual gaps at the perfect-perception extreme (▲) reflect the velocity singularity near $\tau = 1$. The near-optimality of $D_{s,t}$ in the interior confirms that the flow-map trajectory closely approximates the W_2 geodesic even in a non-Gaussian setting, bridging Theorem 1 to the image experiments in Section 3.3.

D.4 Empirical analysis of the PnP iteration

Convergence. We do not provide convergence guarantees for Algorithm 1. Empirically, convergence is enforced by the annealing schedule ($s_k \to 1$ shrinks the denoiser's contribution), not by contraction: the measured local Lipschitz constant of the PnP operator exceeds one and grows with t (see below).

Below, we characterize the behavior of the iteration empirically. To measure regularity, we consider linear inverse problems $f(\mathbf{x}) = \mathbf{H}\mathbf{x}$ and the deterministic variant of the iteration at a fixed noise level s (i.e., $\boldsymbol{\epsilon} = \mathbf{0}$), in which the renoising step reduces to $R_s(\mathbf{z}) = s\,\mathbf{z}$ and one iteration is the operator

$$T_t(\mathbf{x}) := D_{s,t}\big(R_s(F_\lambda(\mathbf{x}))\big), \quad \text{with} \quad F_\lambda(\mathbf{x}) = \mathbf{x} - \lambda \nabla g(\mathbf{x}), \quad R_s(\mathbf{z}) = s\,\mathbf{z}. \tag{36}$$

Let $L_D(t)$ denote the Lipschitz constant of $D_{s,t}$ and $L_F = \|\mathbf{I} - \lambda\,\mathbf{H}^\top\mathbf{H}\|$ that of F_λ, so that T_t is $s\,L_D(t)\,L_F$-Lipschitz. The gradient step satisfies $L_F \le 1$ for $\lambda \in (0,\, 2/\|\mathbf{H}^\top\mathbf{H}\|]$, with $L_F = 1$ when $\mathbf{H}^\top\mathbf{H}$ is singular, as in inpainting.

The composite bound exceeds one. Fig. 38 reports $s\,L_D(t)\,L_F$ on CelebA 128×128 for Gaussian deblurring (kernel size 45, standard deviation 3.0, $\lambda = 1.0$, giving $L_F = 1.000$). The denoiser constant $L_D(t)$ is estimated by power iteration on the Jacobian of $D_{s,t}$ at noisy samples $\mathbf{x}_s = s\,\mathbf{x}_1 + (1-s)\,\boldsymbol{\epsilon}$, taking the worst case over 20 samples. Across all (s, t) tested, the bound exceeds 1 and increases with t, so T_t is not a global contraction.

Local regularity at the iterates. We also measure the local Lipschitz constant $\hat{\rho} = \sigma_{\max}(\partial T_t/\partial \mathbf{x})$ of the full composite operator by power iteration on its Jacobian, evaluated at noisy samples $\mathbf{x}_s =$

$s\,\mathbf{x}_1 + (1-s)\,\boldsymbol{\epsilon}$ and at the actual iterates of the algorithm (iterations $k = 5$ and $k = 250$, 8 images), for Gaussian deblurring and for a well-conditioned control ($\mathbf{H} = \mathbf{I}$, $\lambda = 0.5$); see Fig. 37. We observe that:

(i) at noisy samples, $\hat{\rho}$ increases monotonically with the lookahead at every noise level; e.g., at $s = 0.5$, $\hat{\rho}$ grows from 1.35 ($t = s$) to 1.85, 2.64, and 3.17 ($t = 1$);

(ii) at all operating points, $\hat{\rho} \in [1.3, 9.5]$ at the iterates, including the well-conditioned control, so the iterates do not stabilize because of local contraction; correspondingly, the fixed-s iteration does not converge (the ratio of successive residuals is 1.000 ± 10^{-4}); at the iterates, $\hat{\rho}$ is also larger at $t = 1$ than at $t = s$ in five of six settings, but not monotone in between;

(iii) the deployed algorithm, with the annealed schedule $s_k = k/N$, stabilizes approximately geometrically, with a ratio of successive residuals between 0.958 and 0.962 that does not depend on t.

Together, these measurements indicate that the stability of Algorithm 1 is enforced by the annealing schedule, since the correction applied by the denoiser vanishes as $s_k \to 1$, and not by contractivity of the denoiser. Global convergence of PnP without contractivity can be established under other assumptions, such as averagedness (Pesquet et al., 2021) or gradient-step denoisers in a non-convex optimization framework (Hurault et al., 2022); extending such analyses to flow-map denoisers is left for future work. The growth of $\hat{\rho}$ with t is also consistent with the asymmetry between the two endpoints observed in Appendix C.12.

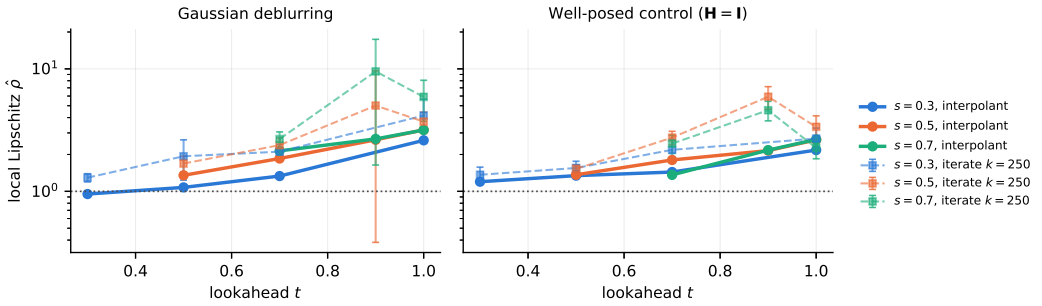


Figure 37: Local Lipschitz constant $\hat{\rho} = \sigma_{\max}(\partial T_t/\partial \mathbf{x})$ of the deterministic PnP operator on CelebA, as a function of the lookahead t, for three noise levels s (log scale; mean over 8 images). Solid: at noisy samples $\mathbf{x}_s$; dashed: at the iterates of the fixed-s iteration ($k = 250$, error bars: one standard deviation). The dotted line marks $\hat{\rho} = 1$.

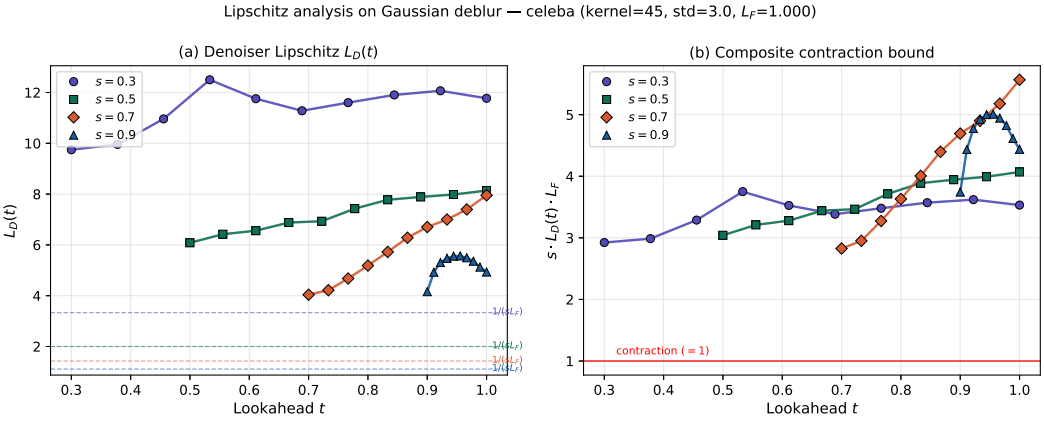


Figure 38: Composite bound $s\,L_D(t)\,L_F$ for the deterministic PnP operator on CelebA Gaussian deblurring. The bound exceeds 1 for all (s, t), so the operator is not a global contraction. $L_D(t)$ is estimated via power iteration on $\partial D_{s,t}/\partial \mathbf{x}$ at noisy samples (worst case over 20). The bound increases with t, with the steepest growth at low noise (large s).

D.5 Gaussian linear inverse problems: fixed point of the iteration

What does the iteration target? PnP is derived from proximal splitting and targets a MAP solution, and plugging in our denoiser does not change this: the lookahead does not switch between optimization and posterior sampling, but changes the implicit regularizer, as a denoiser is the proximal operator of a smoothed version of the regularizer rather than of $-\log p$ itself (Gribonval and Nikolova, 2021). In the Gaussian linear case, the fixed point is a MAP estimator under a surrogate Gaussian prior whose variance grows with t (see below); beyond this case we have no closed form, and we state the resulting behavior as an empirical observation.

We now extend the analysis of Section 3.2.1, where the denoiser was treated as a standalone estimator, to linear inverse problems with an isotropic Gaussian prior $p_1 = \mathcal{N}(\mathbf{0},\, \sigma_p^2\, \mathbf{I})$, in order to characterize *what* the deterministic iteration (36) computes. In this case, the average denoiser is the linear map $D_{s,t}(\mathbf{x}) = \Lambda(s,t)\, \mathbf{x}$, with the scalar gain (9), and the operator T_t is affine,

$$T_t(\mathbf{x}) = m(t)\, \big(\mathbf{x} - \lambda \mathbf{H}^\top (\mathbf{H}\mathbf{x} - \mathbf{y})\big), \qquad m(t) := s\, \Lambda(s,t). \tag{37}$$

Its fixed point admits the closed form

$$\mathbf{x}^\star(t) \;=\; \big(c(t)\, \mathbf{I} + \lambda\, \mathbf{H}^\top \mathbf{H}\big)^{-1}\, \lambda\, \mathbf{H}^\top \mathbf{y}, \qquad c(t) \;=\; \frac{1}{m(t)} - 1, \tag{38}$$

i.e., a Tikhonov-regularized estimator whose regularization weight $c(t)$ depends on the lookahead. We have $m(t) \in (0,1)$, so $c(t) > 0$; since $\Lambda(s,t)$ is strictly increasing in t (Lemma 2), $c(t)$ is strictly decreasing, so the regularization progressively weakens as the lookahead increases.

Interpretation as a MAP estimator. Equivalently, $\mathbf{x}^\star(t)$ minimizes $\frac{1}{2\sigma_{\text{obs}}^2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|^2 + \frac{1}{2\tau^2(t)}\|\mathbf{x}\|^2$ with $\tau^2(t) = \lambda\sigma_{\text{obs}}^2/c(t)$, where σ_{obs}^2 is the measurement noise variance. It is therefore the MAP estimator under a *surrogate* Gaussian prior $\mathcal{N}(\mathbf{0}, \tau^2(t)\mathbf{I})$, whose variance grows with t. It does not coincide with the MAP (Wiener) estimator under the true prior, which would require $c(t)/\lambda = \sigma_{\text{obs}}^2/\sigma_p^2$: the flow-map denoiser operates at the interpolant noise level σ_s of $\mathbf{x}_s$ rather than at the measurement noise level, so it is not the proximal operator of $-\log p_1$ with the corresponding weight. This is an instance of the general fact that a denoiser is the proximal operator of a smoothed regularizer rather than of the negative log-prior itself (Gribonval and Nikolova, 2021). Hence, in this setting, the lookahead does not move the output away from a MAP-type solution; it changes the implicit regularizer, and thus acts as a regularization parameter.

D.5.1 Proof of Equation (38)

By definition, the fixed point $\mathbf{x}^\star(t)$ satisfies $\mathbf{x}^\star(t) = T_t(\mathbf{x}^\star(t))$, i.e.,

$$\mathbf{x}^\star(t) = m(t)\mathbf{x}^\star(t) - m(t)\lambda\mathbf{H}^\top\mathbf{H}\mathbf{x}^\star(t) + m(t)\lambda\mathbf{H}^\top\mathbf{y}.$$

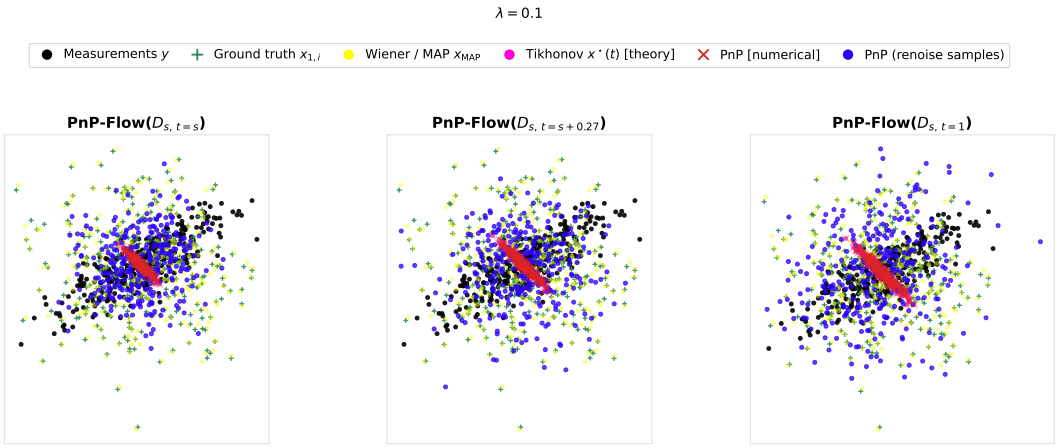
Since $m(t) > 0$, dividing by $m(t)$ and rearranging gives

$$\left[\left(\frac{1}{m(t)} - 1\right)\mathbf{I} + \lambda\mathbf{H}^\top\mathbf{H}\right]\mathbf{x}^\star(t) = \lambda\mathbf{H}^\top\mathbf{y}.$$

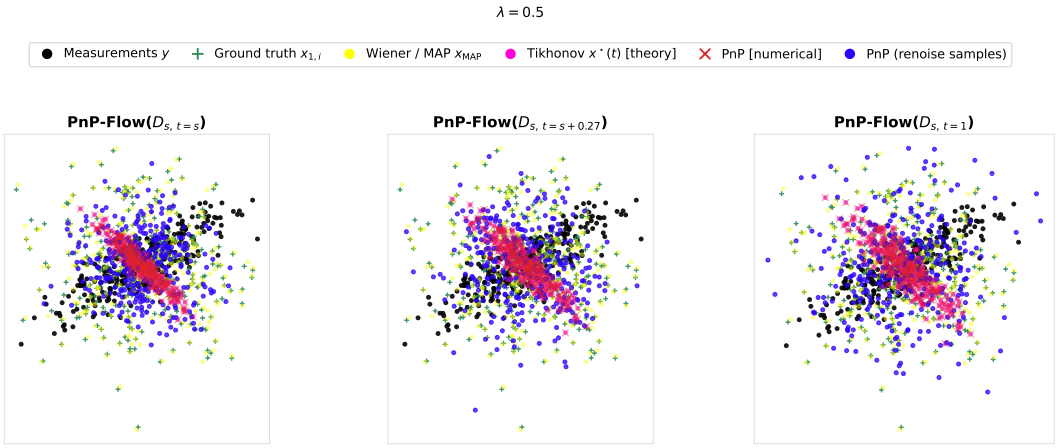
The matrix on the left-hand side is positive definite because $c(t) = 1/m(t) - 1 > 0$, and inverting it yields (38).

Empirical validation. To corroborate the closed form (38), we consider a 2D Gaussian prior $p_1 = \mathcal{N}(\mathbf{0}, \sigma_p^2\,\mathbf{I})$ with $\sigma_p = 2$, and two linear forward models: an isotropic rotation $\mathbf{H}_1 = 0.8\, R(\pi/6)$ and an anisotropic operator $\mathbf{H}_2 = R(\pi/6)\,\mathrm{diag}(1.0,\, 0.3)\, R(\pi/4)$, whose Gram matrix $\mathbf{H}_2^\top\mathbf{H}_2$ has eigenvalues $\{1.0,\, 0.09\}$ and couples the coordinates. Observations are $\mathbf{y} = \mathbf{H}_k\, \mathbf{x}_1 + \eta$ with $\eta \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{obs}}^2\, \mathbf{I})$, $\sigma_{\text{obs}} = 0.1$.

We run deterministic PnP-Flow ($\boldsymbol{\epsilon} = \mathbf{0}$) with $\lambda = 0.1$ and 0.5, anchor $s = 0.45$, and lookaheads $t \in \{s,\, s+0.27,\, 1\}$. The iterates converge to $\mathbf{x}^\star(t)$ to machine precision ($\|\hat{\mathbf{x}} - \mathbf{x}^\star(t)\|_{\text{mean}} < 5\cdot 10^{-16}$), validating (38). As t increases, the effective regularization $c(t)/\lambda$ decreases ($0.93 \to 0.59 \to 0.43$), yielding a continuum of Tikhonov estimators with progressively weaker shrinkage toward the prior mean. The result is shown in Fig. 39.



(a)



(b)

Figure 39: Two forward models are tested: **(a)** isotropic rotation $\mathbf{H}_1 = 0.8\, R(\pi/6)$ with $\lambda = 0.1$, and **(b)** anisotropic operator $\mathbf{H}_2 = R(\pi/6)\,\mathrm{diag}(1.0, 0.3)\,R(\pi/4)$ with $\lambda = 0.5$. Black: measurements $\mathbf{y}$. Gold crosses: ground truth $\mathbf{x}_{1,i}$. Yellow: Wiener/MAP estimator. Red dots: Tikhonov fixed point $\mathbf{x}^\star(t)$ predicted by Eq. (38). Red $\times$: PnP iterates after convergence (matching theory to machine precision, $\|\hat{\mathbf{x}} - \mathbf{x}^\star(t)\|_{\mathrm{mean}} < 5 \cdot 10^{-16}$). Blue dots: samples from the stochastic PnP variant, illustrating the variance restoration discussed in Section 3.3. As t increases (left to right), the Tikhonov regularization weakens and the iterates spread further from the prior mean toward the Wiener solution.

In this Gaussian inverse problem, the lookahead acts as a *regularization knob* rather than as a parameterization of the optimal DP curve. Since this regime is over-regularized ($c(t)/\lambda$ exceeds $\sigma^2_{\mathrm{obs}}/\sigma^2_p$), increasing t reduces bias and moves $\mathbf{x}^\star(t)$ toward the Wiener solution, which also decreases the MSE. This does not contradict the image experiments: exact DP optimality holds only in the denoising setting, while in inverse problems the interaction with the forward model breaks this guarantee.

Effect of renoising in the Gaussian setting. To isolate the role of the renoising step, we repeat the experiment with the full stochastic PnP iteration. While the deterministic case converges to a single Tikhonov solution, the stochastic iteration produces a distribution of estimates whose spread is largely independent of the step size λ.

Across different λ, the sample clouds consistently exhibit meaningful variance and align with the geometry of the data distribution, even when the mean estimate remains over-regularized. This indicates that the renoising step, together with the denoiser, primarily controls the variance of the estimator, while λ governs its mean through the data-consistency update.

As the lookahead increases, this variability is progressively amplified, yielding the distortion–perception behavior observed in the experiment.

E Broader impacts

Image restoration methods that generate content under measurement constraints can produce plausible but incorrect details, particularly at the perceptual end of the DP tradeoff and for degradations with large null spaces, such as inpainting. In sensitive applications, such as medical imaging, forensics, or security, such hallucinations could lead to wrong decisions. The lookahead studied in this work makes this tradeoff explicit and controllable, since users can select a low-distortion operating point when fidelity is critical, but reconstructions should still be validated before being used in high-stakes settings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the three regimes of our claims: exact DP optimality for Gaussian denoising with isotropic targets (Theorem 1), empirical DP traversal for natural-image denoising (Section 3.3), and empirical DP traversal without optimality guarantees for general inverse problems (Sections 4 and 6). We support this claim with an extensive ablation and comparison with baselines.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 7 and Appendix C.13, including the scope of the Gaussian result (Section 3.2.1, Appendix D.2.3), the absence of convergence guarantees (Appendix D.4), non-monotone DP behavior, and training cost (Appendix B.1).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 1 states its assumptions, and its complete proof, together with the supporting lemmas and the multivariate discussion, is in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our algorithm is clearly described in Alg. 1, and all the hyperparameters and datasets used are explicitly reported (see Appendix B.2). The source code is also provided in the submission (it will be publicly available on GitHub if the paper gets accepted).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code to reproduce the experiments is publicly available at `https://github.com/nzilberstein/Flow-map-denoisers`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, see Appendix B.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, in Appendix C.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we provide the information used for sampling and training in Appendix B.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and ensured the paper satisfies every aspect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: See Appendix E.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We used well-established datasets that are curated, and we validate that all our trained models satisfy the safeguard requirements.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit all external code assets used in our work by citing the original papers. The only assets that are not ours and were used in this work are the backbone architecture (credited when citing the original paper by Karras et al. (2022); Song et al. (2021)), and the public datasets (credited in Appendix B.1).

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All the information regarding model and datasets is provided in Appendix B.2.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We did not do experiments with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We did not do experiments with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our method does not involve LLMs in any of the core modules.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.