FIRM: FLOW-BASED IMAGING VIA REGULARIZED MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching methods for imaging inverse problems typically incorporate measurements through network conditioning or guidance during sampling. Neither approach explicitly applies the forward operator within the learned conditional velocity field. We develop a principled measurement-conditional velocity parameterization that does. For a linear interpolation path, we express the optimal velocity through the posterior mean $\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{x}_t, \boldsymbol{y}]$ and show that this mean is the unique minimizer of a variational objective with an explicit data-consistency term. The velocity defined by this minimizer provably transports the source distribution to the measurement-conditioned posterior. This result leads to a forward operator-aware velocity field that is trained end-to-end and requires no separate guidance during sampling. Across five imaging tasks, our method achieves leading reconstruction quality with up to $50\times$ fewer network evaluations than competitive flow-based methods. Varying the number of sampling steps also controls the distortion-perception trade-off without retraining.

1 INTRODUCTION

Imaging inverse problems aim to recover an unknown image from degraded measurements. They arise in magnetic resonance imaging (MRI), computed tomography (CT), microscopy, astronomical imaging, and remote sensing (Bertero et al., 2021; McCann et al., 2017; Ongie et al., 2020). Because the forward operator is often underdetermined, the measurements do not uniquely determine the image. Reconstruction therefore requires prior information. A common classical approach combines a data-fidelity term with a hand-designed regularizer, such as sparsity or smoothness (Tikhonov & Arsenin, 1977; Vogel, 2002; Tarantola, 2005; Mohammad-Djafari, 2021). Learning-based methods often replace or complement hand-designed regularization with priors learned from data.

Score-based diffusion models (Ho et al., 2020; Song et al., 2021) and flow-matching models (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023b) provide expressive learned priors for imaging inverse problems (Daras et al., 2024). A key distinction among generative inverse-problem solvers is how and when measurement information enters the reconstruction.

A common approach starts with a pretrained image prior $p(\boldsymbol{x})$ and introduces measurements during inference. DiffPIR (Zhu et al., 2023) and DDRM (Kawar et al., 2022) use proximal or spectral data-consistency updates, while variational PnP diffusion (Mardani et al., 2024) alternates denoising with data-consistency projections. Flow-based methods similarly combine an unconditional velocity field with gradient- or projection-based consistency updates (Martin et al., 2025; Pourya et al., 2026; Zhang et al., 2024). Reusing a prior across forward models is valuable, but the resulting solver must repeatedly correct its generative trajectory using measurements at inference, often requiring many velocity evaluations to produce a reconstruction. See Appendix C for further discussion.

Another approach trains a score or velocity field conditioned on the measurement $\boldsymbol{y}$ (Saharia et al., 2022b; Delbracio & Milanfar, 2023; Liu et al., 2023a). Conditioning can eliminate separate guidance at inference, but measurement information typically enters only as a network input, such as $\boldsymbol{y}$, $\boldsymbol{A}^\mathsf{T}\boldsymbol{y}$, or a learned embedding. Data consistency must therefore be learned from paired examples rather than imposed through an explicit forward operator-dependent update.

We give explicit data consistency a different role: it becomes part of each evaluation of a learned measurement-conditional velocity field. Under linear interpolation, $\boldsymbol{x}_t$ and $\boldsymbol{y}$ are conditionally in-

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

dependent noisy observations of the clean image $\boldsymbol{x}_1$, and their joint posterior mean determines the optimal conditional velocity, $\boldsymbol{v}(\boldsymbol{x}_t, t, \boldsymbol{y}) = \big(\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{x}_t, \boldsymbol{y}] - \boldsymbol{x}_t\big)/(1-t)$. We characterize this posterior mean as the unique minimizer of a variational objective with interpolation consistency, explicity measurement consistency, and an implicit regularizer. We then prove that the velocity defined by this minimizer transports the source distribution to the measurement-conditioned posterior. Guided by this characterization, we alternate a forward operator-dependent data-consistency steps with a learned update to parametrize the velocity field. The full parameterization is trained end-to-end by flow matching and requires no separate guidance during sampling.

Our contributions are threefold. **(a)** We introduce FIRM, a novel measurement-conditional flow-matching method that explicitly incorporates data-consistency into each velocity field evaluation. **(b)** We characterize the optimal conditional velocity through a variational minimizer and show that the resulting exact field transports the source distribution to the measurement-conditioned posterior. **(c)** Across five imaging tasks, our method achieves leading reconstruction quality with up to $50\times$ fewer network evaluations than competitive flow-based methods. Varying the number of sampling steps also controls the distortion-perception trade-off without retraining.

2 Background

Inverse problems. Imaging inverse problems seek to recover an unknown image $\boldsymbol{x} \in \mathbb{R}^n$ from degraded measurements $\boldsymbol{y} \in \mathbb{R}^m$, commonly modeled as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{A} : \mathbb{R}^n \to \mathbb{R}^m$ is the measurement operator and $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is additive Gaussian noise with standard deviation σ. Because $\boldsymbol{A}$ may be noninvertible or underdetermined, the measurements may not uniquely determine $\boldsymbol{x}$; even when the solution is unique, ill-conditioning can make recovery unstable. Thus, reconstruction requires prior information. A common variational estimator is

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{with} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + h(\boldsymbol{x}), \tag{2}$$

where $g(\boldsymbol{x})$ enforces measurement consistency and $h(\boldsymbol{x})$ encodes prior information. Under additive Gaussian noise, a standard choice is $g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$, while h may promote properties such as total variation, wavelet sparsity, or sparse dictionary representations (Rudin et al., 1992; Beck & Teboulle, 2009; Figueiredo & Nowak, 2003).

PnP and HQS. Plug-and-Play (PnP) methods retain the data-fidelity term and replace the proximal update associated with a hand-designed regularizer by a learned denoiser (Venkatakrishnan et al., 2013). The proximal operator is

$$\mathrm{prox}_{\gamma h}(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}} \left\{ h(\boldsymbol{u}) + \frac{1}{2\gamma}\|\boldsymbol{u} - \boldsymbol{x}\|_2^2 \right\}. \tag{3}$$

When h is a negative log-prior, this operator corresponds to MAP denoising with Gaussian noise level $\sqrt{\gamma}$. PnP methods instead use a learned denoiser to encode prior information implicitly (Romano et al., 2017; Ahmad et al., 2020; Ryu et al., 2019; Kamilov et al., 2023). Half-quadratic splitting (HQS) introduces an auxiliary variable $\boldsymbol{z}$ and separates the data-fidelity and regularization terms as

$$\min_{\boldsymbol{x},\boldsymbol{z}} \left\{ g(\boldsymbol{x}) + h(\boldsymbol{z}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 \right\}, \tag{4}$$

where $\mu > 0$ controls the coupling between $\boldsymbol{x}$ and $\boldsymbol{z}$. Alternately minimizing gives

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}} \left\{ g(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{z}^k\|_2^2 \right\} = \left(\boldsymbol{A}^\top \boldsymbol{A} + \mu \boldsymbol{I}\right)^{-1} \left(\boldsymbol{A}^\top \boldsymbol{y} + \mu \boldsymbol{z}^k\right), \tag{5}$$

$$\boldsymbol{z}^{k+1} = \arg\min_{\boldsymbol{z}} \left\{ h(\boldsymbol{z}) + \frac{\mu}{2}\|\boldsymbol{x}^{k+1} - \boldsymbol{z}\|_2^2 \right\} = \mathrm{prox}_{h/\mu}\left(\boldsymbol{x}^{k+1}\right). \tag{6}$$

The $\boldsymbol{x}$-update enforces data consistency, while the $\boldsymbol{z}$-update applies prior information. PnP replaces the latter with a learned denoiser at noise level $1/\sqrt{\mu}$. Algorithm unrolling treats the alternating updates as differentiable network layers that can be trained end-to-end (Gregor & LeCun, 2010; Monga et al., 2021).

Flow matching. Flow matching (Lipman et al., 2023; Liu et al., 2023b; Albergo & Vanden-Eijnden, 2023) constructs generative models by learning continuous transport dynamics from a source distribution p_0 to a target distribution p_1. The intermediate distributions $\{p_t\}_{t\in[0,1]}$ satisfy the continuity equation

$$\partial_t p_t + \nabla \cdot (p_t \boldsymbol{v}_t) = 0, \tag{7}$$

where $\boldsymbol{v}_t$ is the velocity field and samples evolve according to $d\boldsymbol{x}_t/dt = \boldsymbol{v}_t(\boldsymbol{x}_t)$. Following the linear interpolation path used in flow matching and rectified flows (Lipman et al., 2023; Liu et al., 2023b), we have

$$\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1, \qquad t \in [0,1], \tag{8}$$

where $\boldsymbol{x}_0 \sim p_0$ and $\boldsymbol{x}_1 \sim p_1$. The optimal marginal velocity is $\boldsymbol{v}(\boldsymbol{x}_t, t) = \mathbb{E}[\boldsymbol{x}_1 - \boldsymbol{x}_0 | \boldsymbol{x}_t]$. Because this conditional expectation is generally unavailable, conditional flow matching (CFM) (Lipman et al., 2023; Tong et al., 2024) regresses directly onto the sample-wise path velocity,

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_1}\left[\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|^2\right], \tag{9}$$

whose gradient matches that of the marginal flow-matching objective. Here, "conditional" refers to the probability path used to construct the training target, not to measurement conditioning. For the linear path, the optimal velocity can also be written as

$$\boldsymbol{v}(\boldsymbol{x}_t, t) = \mathbb{E}\left[\boldsymbol{x}_1 - \boldsymbol{x}_0 | \boldsymbol{x}_t\right] = \frac{\mathbb{E}\left[\boldsymbol{x}_1 | \boldsymbol{x}_t\right] - \boldsymbol{x}_t}{1-t}. \tag{10}$$

Thus, velocity estimation reduces to conditional-mean estimation. The sample-wise paths are straight, motivating numerical integration with relatively few steps in rectified-flow models (Liu et al., 2023b; Tong et al., 2024). After training, samples are generated by solving

$$\hat{\boldsymbol{x}}_1 = \boldsymbol{x}_0 + \int_0^1 \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\,\mathrm{d}t. \tag{11}$$

toward $t = 1$. Numerical solvers evaluate the representation in Eq. (10) only for $t < 1$, where it remains well defined.

3 Method

3.1 Theoretical Characterization

Under linear interpolation, the interpolation state $\boldsymbol{x}_t$ and measurement $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{\eta}$ are conditionally independent noisy observations of $\boldsymbol{x}_1$. They form the augmented observation model

$$\underbrace{\begin{bmatrix}\boldsymbol{x}_t \\ \boldsymbol{y}\end{bmatrix}}_{\widetilde{\boldsymbol{y}}} = \underbrace{\begin{bmatrix}t\boldsymbol{I} \\ \boldsymbol{A}\end{bmatrix}}_{\widetilde{\boldsymbol{A}}} \boldsymbol{x}_1 + \underbrace{\begin{bmatrix}(1-t)\boldsymbol{x}_0 \\ \boldsymbol{\eta}\end{bmatrix}}_{\widetilde{\boldsymbol{b}}}, \qquad \widetilde{\boldsymbol{b}} \sim \mathcal{N}\left(\boldsymbol{0}, \mathrm{diag}\left((1-t)^2\boldsymbol{I}, \sigma^2\boldsymbol{I}\right)\right). \tag{12}$$

The MMSE estimator of $\boldsymbol{x}_1$ under the augmented observation model ($\widetilde{\boldsymbol{y}} = \widetilde{\boldsymbol{A}}\boldsymbol{x}_1 + \widetilde{\boldsymbol{b}}$) is the posterior mean $\mathbb{E}[\boldsymbol{x}_1 | \boldsymbol{x}_t, \boldsymbol{y}]$. The following proposition characterizes the posterior mean as the minimizer of a variational objective and shows that the velocity it defines transports the source distribution to the measurement-conditioned posterior.

Proposition 1. *For a non-degenerate p_1 with bounded support, $\sigma > 0$, and $t \in (0,1)$:*

(a) There exists a regularizer $\phi_t : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, unique up to an additive constant, such that $\mathbb{E}[\boldsymbol{x}_1 | \boldsymbol{x}_t, \boldsymbol{y}]$ is the unique global minimizer, and the unique stationary point of

$$\mathbb{E}[\boldsymbol{x}_1 | \boldsymbol{x}_t, \boldsymbol{y}] = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \frac{1}{2(1-t)^2}\|\boldsymbol{x}_t - t\boldsymbol{x}\|_2^2 + \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \phi_t(\boldsymbol{x}), \tag{13}$$

where the minimizer determines the optimal measurement-conditional velocity,

$$\boldsymbol{v}(\boldsymbol{x}_t, t, \boldsymbol{y}) = \frac{\mathbb{E}[\boldsymbol{x}_1 | \boldsymbol{x}_t, \boldsymbol{y}] - \boldsymbol{x}_t}{1-t}. \tag{14}$$

(b) Moreover, $\boldsymbol{v}$ satisfies the conditional continuity equation

$$\partial_t p_t(\boldsymbol{x} \mid \boldsymbol{y}) + \nabla \cdot (p_t(\boldsymbol{x} \mid \boldsymbol{y}) \boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{y})) = 0. \tag{15}$$

For every $\tau < 1$, the corresponding flow transports the Gaussian source to $p_\tau(\cdot \mid \boldsymbol{y})$, and $p_\tau(\cdot \mid \boldsymbol{y}) \Rightarrow p(\boldsymbol{x}_1 \mid \boldsymbol{y})$ as $\tau \to 1^-$.

The proof is provided in Appendix A.3. The variational statement applies the MMSE characterization of Gribonval & Machart (2013) to the augmented model in Eq. (12). In Eq. (13), two quadratic terms enforce interpolation and measurement consistency, while ϕ_t captures the remaining structure. The minimizer yields the posterior mean and hence the optimal conditional velocity, whose probability flow transports the source distribution to the measurement-conditioned posterior. Because ϕ_t is unavailable, the next subsection turns this principled characterization into a practical algorithm.

3.2 Operator-Aware Conditional Velocity

Proposition 1 reduces the construction of the optimal conditional velocity to minimizing Eq. (13). However, the implicit regularizer ϕ_t is not available in closed form. We therefore apply HQS to separate the objective into an explicit data-consistency update and a learned update associated with ϕ_t. Introducing an auxiliary variable $\boldsymbol{z}$ and a quadratic penalty gives

$$\min_{\boldsymbol{x},\boldsymbol{z}} \frac{1}{2(1-t)^2}\|\boldsymbol{x}_t - t\boldsymbol{z}\|_2^2 + \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \phi_t(\boldsymbol{z}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2, \tag{16}$$

where $\mu > 0$ controls the coupling. Alternating minimization over $\boldsymbol{x}$ and $\boldsymbol{z}$ yields

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}} \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{z}^k\|_2^2, \tag{17}$$

$$\boldsymbol{z}^{k+1} = \arg\min_{\boldsymbol{z}} \phi_t(\boldsymbol{z}) + \frac{1}{2(1-t)^2}\|\boldsymbol{x}_t - t\boldsymbol{z}\|_2^2 + \frac{\mu}{2}\|\boldsymbol{x}^{k+1} - \boldsymbol{z}\|_2^2. \tag{18}$$

The $\boldsymbol{x}$-update explicitly enforces consistency with the measurement model, while the $\boldsymbol{z}$-update combines interpolation consistency with implicit regularization. Because ϕ_t is unknown, we implement the $\boldsymbol{z}$-update using a learned estimator. The complete implementation of the algorithm is provided in Appendix B and Algorithm 3.

Let $\boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_t, t, \boldsymbol{y})$ denote the output after K iterations. The output estimates the measurement-conditioned posterior mean $\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{x}_t, \boldsymbol{y}]$ and defines the learned velocity

$$\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{y}) = \frac{\boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_t, t, \boldsymbol{y}) - \boldsymbol{x}_t}{1-t}. \tag{19}$$

The forward operator therefore enters the learned velocity through the data-consistency update at every iteration.

3.3 Training the Conditional Velocity Field

Given a clean image $\boldsymbol{x}_1$, we generate a measurement $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{\eta}$, sample $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ independently, and draw $t \sim \mathcal{U}[t_{\min}, t_{\max}]$. The interpolation state is

$$\boldsymbol{x}_t = t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0. \tag{20}$$

Algorithm 3 produces the operator-aware estimate $\boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_t, t, \boldsymbol{y})$ of the measurement-conditioned posterior mean. We use the stabilized time scale

$$\tau_t = \max(1 - t, \tau_{\min}), \tag{21}$$

where $t_{\max} < 1$ excludes the singular endpoint and $\tau_{\min} > 0$ bounds the velocity scaling. The predicted and target velocities are

$$\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{y}) = \frac{\boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_t, t, \boldsymbol{y}) - \boldsymbol{x}_t}{\tau_t}, \qquad \boldsymbol{v}_{\text{target}} = \frac{\boldsymbol{x}_1 - \boldsymbol{x}_t}{\tau_t}, \tag{22}$$

where $\boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_t, t, \boldsymbol{y})$ is the posterior mean provided by Algorithm 3. We train the complete estimator end-to-end with the weighted flow-matching loss

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_0, t, \boldsymbol{y}}\left[\omega(t)\,\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{y}) - \boldsymbol{v}_{\text{target}}\|_2^2\right], \qquad \omega(t) = \frac{1 + \tau_t^3}{\tau_t}. \tag{23}$$

Algorithm 1 summarizes the training procedure.

Algorithm 1 FIRM Training

Require: $\boldsymbol{A}$, σ, $[t_{\min}, t_{\max}]$, and $\tau_{\min}$
1: **repeat**
2: $\boldsymbol{x}_1 \sim p_1$, $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $t \sim \mathcal{U}[t_{\min}, t_{\max}]$
3: $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$, $\boldsymbol{y} \leftarrow \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{\eta}$
4: $\boldsymbol{x}_t \leftarrow t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0$, $\tau_t = \max(1-t, \tau_{\min})$
5: $\boldsymbol{z}_{\boldsymbol{\theta}}^K \leftarrow \textsc{PostMean}(\boldsymbol{x}_t, t, \boldsymbol{y}, \boldsymbol{A})$ ▷ Alg. 3
6: $\boldsymbol{v}_{\boldsymbol{\theta}} \leftarrow (\boldsymbol{z}_{\boldsymbol{\theta}}^K - \boldsymbol{x}_t)/\tau_t$
7: $\boldsymbol{v}_{\text{target}} \leftarrow (\boldsymbol{x}_1 - \boldsymbol{x}_t)/\tau_t$
8: Update $\boldsymbol{\theta}$ using Eq. (23)
9: **until** convergence

Algorithm 2 FIRM Sampling

Require: Measurement $\boldsymbol{y}$, forward operator $\boldsymbol{A}$, trained model, and step count N
1: Sample $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
2: $\Delta t \leftarrow 1/N$
3: **for** $i = 0, \ldots, N-1$ **do**
4: $t_i \leftarrow i\Delta t$
5: $\boldsymbol{z}_{\boldsymbol{\theta},i}^K \leftarrow \textsc{PostMean}(\boldsymbol{x}_i, t_i, \boldsymbol{y}, \boldsymbol{A})$ ▷ Alg. 3
6: $\boldsymbol{v}_i \leftarrow (\boldsymbol{z}_{\boldsymbol{\theta},i}^K - \boldsymbol{x}_i)/(1-t_i)$
7: $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \Delta t\, \boldsymbol{v}_i$
8: **return** $\boldsymbol{x}_N$

Table 1: Quantitative results on 100 CelebA test images. Best and second-best results are highlighted in blue and red, respectively. FIRM achieves the best reconstruction quality while attaining competitive perceptual quality.

Method	Denoising			Deblurring			Super-resolution			Random inpainting			Box inpainting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Degraded	20.00	0.348	0.371	27.81	0.740	0.125	10.25	0.183	0.827	11.95	0.196	1.041	22.26	0.743	0.213
PnP-GS	32.54	0.908	0.035	33.97	0.924	0.041	31.23	0.890	0.065	29.20	0.875	0.069	–	–	–
PnP-HQS	32.17	0.897	0.044	34.87	0.946	0.033	32.33	0.911	0.075	32.81	0.945	0.023	–	–	–
ISTA-Net	-	-	-	35.38	0.950	0.035	33.54	0.937	0.038	34.70	0.960	0.022	–	–	–
DiffPIR	31.12	0.883	0.060	32.68	0.910	0.060	31.46	0.893	0.034	31.69	0.916	0.025	–	–	–
I^2SB	-	-	-	33.25	0.924	0.012	31.92	0.891	0.014	32.80	0.938	0.011	28.72	0.803	0.042
OT-ODE	30.49	0.858	0.032	32.96	0.920	0.029	31.34	0.903	0.027	28.65	0.870	0.051	29.37	0.919	0.038
D-Flow	26.01	0.606	0.092	31.21	0.854	0.037	30.44	0.843	0.026	33.61	0.942	0.015	30.59	0.898	0.027
Flow-Priors	29.31	0.767	0.135	31.51	0.857	0.056	28.36	0.715	0.100	32.87	0.943	0.019	30.06	0.859	0.048
PnP-Flow1	31.75	0.904	0.044	34.48	0.936	0.039	31.04	0.901	0.045	33.02	0.944	0.018	30.45	0.933	0.037
PnP-Flow5	32.24	0.910	0.056	34.80	0.941	0.046	31.44	0.905	0.056	33.95	0.953	0.022	31.06	0.939	0.043
Flower1-OT	32.22	0.913	0.033	34.95	0.947	0.025	32.30	0.922	0.034	33.05	0.944	0.017	31.17	0.945	0.022
Flower5-OT	33.07	0.925	0.038	35.65	0.954	0.031	33.03	0.931	0.039	33.92	0.953	0.020	31.85	0.952	0.023
Ours ($N = 2$)	33.67	0.932	0.034	36.04	0.957	0.029	34.39	0.949	0.028	35.03	0.963	0.018	33.25	0.960	0.022
Ours ($N = 4$)	33.24	0.928	0.034	35.78	0.956	0.028	33.96	0.946	0.026	34.67	0.962	0.016	32.57	0.958	0.021
Ours ($N = 10$)	32.26	0.914	0.032	35.03	0.950	0.023	33.13	0.937	0.022	33.98	0.956	0.014	31.85	0.953	0.020
Ours ($N = 25$)	31.56	0.901	0.028	31.46	0.848	0.026	30.39	0.837	0.029	33.42	0.951	0.013	31.33	0.947	0.020

3.4 Sampling with the Conditional Velocity Field

At inference, we use the theoretical scale $1 - t$ rather than the training floor τ_t. For a state $\boldsymbol{x}_t$, Algorithm 3 returns $\boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_t, t, \boldsymbol{y})$, and the learned conditional velocity is computed using Eq. (19). Starting from $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we integrate

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{y}) \tag{24}$$

using N explicit Euler steps. With $\Delta t = 1/N$ and $t_i = i\Delta t$, the update is

$$\boldsymbol{z}_{\boldsymbol{\theta},i}^K = \boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_i, t_i, \boldsymbol{y}), \tag{25}$$

$$\boldsymbol{v}_i = \frac{\boldsymbol{z}_{\boldsymbol{\theta},i}^K - \boldsymbol{x}_i}{1 - t_i}, \tag{26}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta t\, \boldsymbol{v}_i, \qquad i = 0, \ldots, N-1. \tag{27}$$

Since $t_i < 1$ for every velocity evaluation, the denominator remains well-defined. Because the posterior-mean estimator already incorporates the forward operator, sampling requires no external guidance. Changing N provides test-time control over the distortion–perception trade-off without retraining. Algorithm 2 summarizes the sampling procedure.

4 Experimental Results

We evaluate our method on five inverse problems: denoising, deblurring, super-resolution, random-mask inpainting, and box inpainting. Experiments use CelebA (Liu et al., 2015) and AFHQ-Cat (Choi et al., 2020), following the datasets and evaluation protocols of Martin et al. (2025) and Pourya et al. (2026) for a fair comparison.

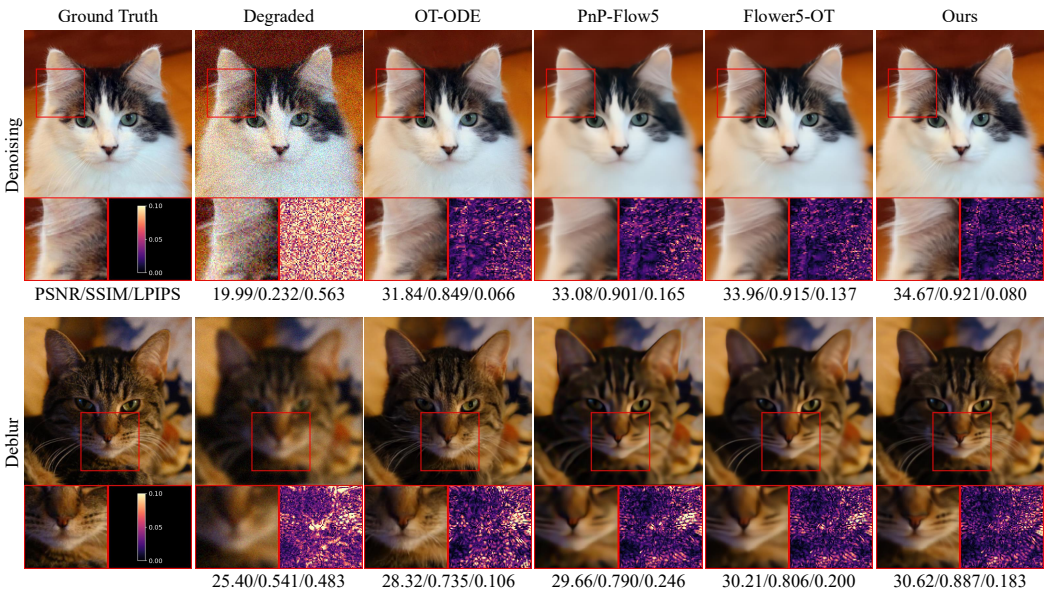

Figure 1: Qualitative reconstructions on AFHQ-Cat for denoising (top) and deblurring (bottom). Values below each panel are PSNR/SSIM/LPIPS. Our method preserves fine image details and produces faithful reconstructions using only $N{=}2$ sampling steps.

CelebA contains aligned face images that we center-crop and resize to 128×128, using the standard training, validation, and test splits. We resize AFHQ-Cat images to 256×256. Because AFHQ-Cat has no official validation split, we follow Martin et al. (2025) and reserve 32 test images for validation. All images are normalized to $[-1, 1]$.

We compare against state-of-the-art PnP, unfolding, diffusion-based, and flow-based inverse solvers: PnP-GS (Hurault et al., 2022), PnP-HQS (DPIR) (Zhang et al., 2021), ISTA-Net (Zhang & Ghanem, 2018), DiffPIR (Zhu et al., 2023), I^2SB (Liu et al., 2023a), OT-ODE (Pokle et al., 2024), D-Flow (Ben-Hamu et al., 2024), Flow Priors (Zhang et al., 2024), PnP-Flow (Martin et al., 2025), and Flower (Pourya et al., 2026). All baselines except ISTA-Net and I^2SB enforce measurement consistency at inference time using a pretrained prior.

4.1 Main Results

We evaluate on five restoration tasks and report PSNR (↑), SSIM (↑), and LPIPS (↓) (Zhang et al., 2018), averaged over 100 test images. We adopt the degradation settings of Martin et al. (2025) and Pourya et al. (2026) without modification. Denoising uses Gaussian noise with $\sigma = 0.2$. Deblurring uses a 61×61 Gaussian kernel, with $\sigma_b = 1.0$ on CelebA and $\sigma_b = 3.0$ on AFHQ-Cat, and $\sigma = 0.05$. Super-resolution uses $2\times$ downsampling on CelebA and $4\times$ on AFHQ-Cat, with $\sigma = 0.05$. Random inpainting masks 70% of pixels with $\sigma = 0.01$, and box inpainting applies a centered mask of 40×40 on CelebA and 80×80 on AFHQ-Cat with $\sigma = 0.05$. In tables, "–" marks settings where a baseline does not apply.

Tables 1 and 2 report results on CelebA and AFHQ-Cat. Our method ranks first or second in most settings. Varying the number of sampling steps controls the distortion-perception trade-off, where fewer steps favor PSNR and SSIM, whereas additional steps improve LPIPS, allowing one trained model to span both regimes without retraining. Figures 1 and 2 show representative reconstructions. Following Pourya et al. (2026), we also evaluate ours and the faster baselines on *larger testsets* in Tables 11 and 12.

4.2 Inference Cost and Scalability

Table 3 reports inference time, peak GPU memory, and number of function evaluations (NFEs) for CelebA deblurring, averaged over 10 images on a single NVIDIA RTX A6000. Our inference cost is

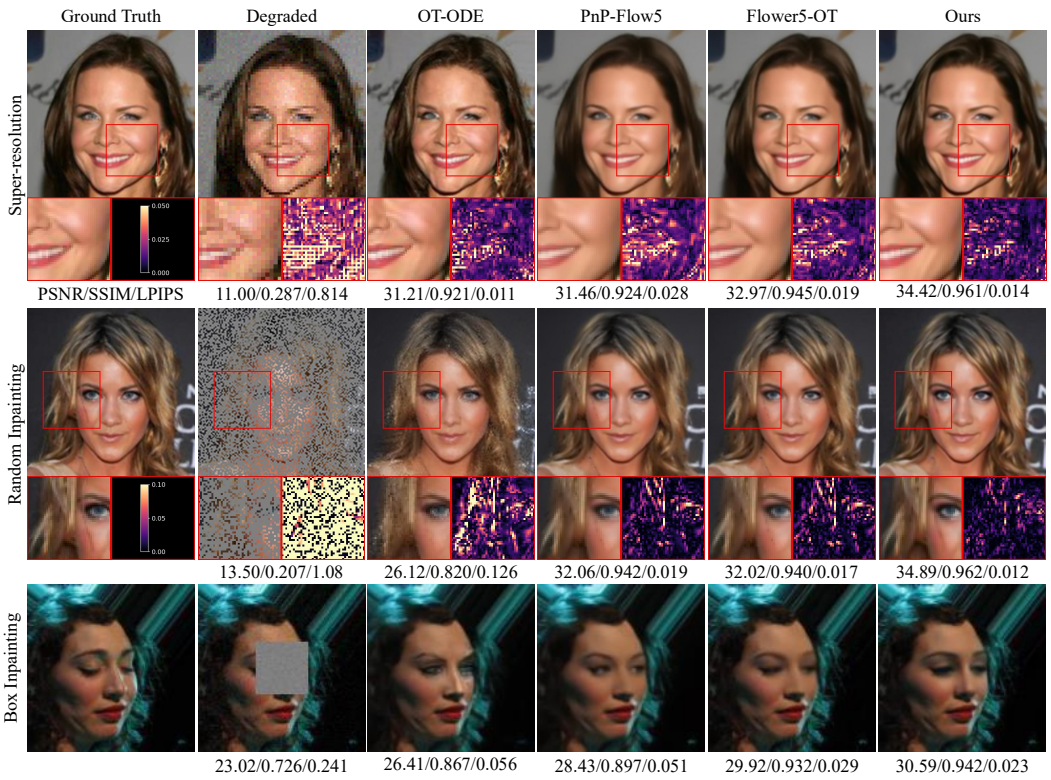

Figure 2: Qualitative reconstructions on CelebA. Values below each panel report PSNR/SSIM/LPIPS. Note that our method recovers details other baselines fail to recover while requiring only $N{=}2$ ODE steps.

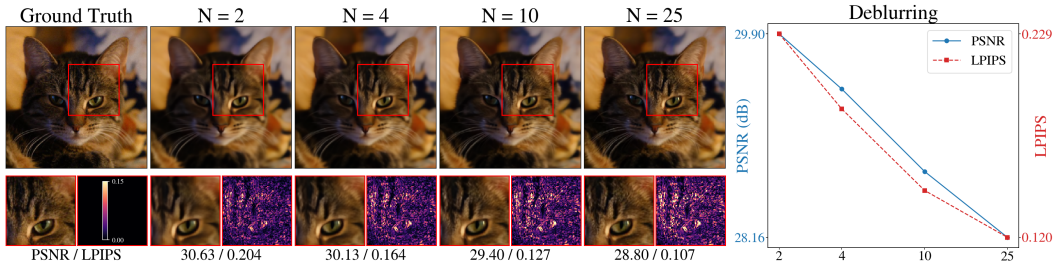

Figure 3: Effect of the number of ODE steps N on AFHQ-Cat deblurring. Values below each reconstruction report PSNR/LPIPS for the displayed image, while the right panel reports averages over the test set. Increasing N improves perceptual quality at the cost of lower PSNR, allowing a single trained model to control the distortion-perception tradeoff at test time.

$O(NK)$, where N is the number of ODE steps and $K{=}5$ is the number of inner HQS iterations; thus, the $N{=}2$ configuration requires 10 NFEs. Although its per-NFE cost is comparable to PnP-Flow and Flower, our method requires $50\times$ fewer evaluations. At the configuration achieving the best distortion metrics, it reconstructs an image in $0.24\,\mathrm{s}$, compared with $13.03\,\mathrm{s}$ for Flower5-OT. This speedup stems from learning data consistency during training rather than enforcing it at inference. Our method uses more peak memory than PnP-Flow and Flower because it retains intermediate iterates, but substantially less than other baselines.

4.3 Ablation Studies

Number of sampling steps N. Figure 3 shows how the number of sampling steps affects reconstruction quality. Across most tasks, increasing N lowers PSNR but improves LPIPS. Coarse integration

Table 2: Quantitative results on 100 AFHQ-Cat test images. Best and second-best results are highlighted in blue and red, respectively. Our method achieves the best reconstruction quality while attaining the best or competitive perceptual quality.

Method	Denoising			Deblurring			Super-resolution			Random inpainting			Box inpainting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Degraded	20.00	0.293	0.529	24.38	0.529	0.436	11.59	0.212	0.867	13.23	0.213	1.082	21.52	0.727	0.215
PnP-GS	33.00	0.895	0.078	28.39	0.787	0.387	24.44	0.639	0.411	29.82	0.844	0.143	-	-	-
PnP-HQS	32.69	0.896	0.108	29.34	0.793	0.362	26.58	0.724	0.441	33.07	0.916	0.051	-	-	-
ISTA-Net	-	-	-	29.36	0.790	0.342	28.22	0.784	0.297	34.10	0.929	0.049	-	-	-
DiffPIR	31.05	0.839	0.186	28.24	0.747	0.319	24.24	0.650	0.385	32.31	0.886	0.057	-	-	-
I^2SB	-	-	-	27.72	0.725	0.077	27.19	0.739	0.070	31.79	0.879	0.026	27.32	0.763	0.072
OT-ODE	30.48	0.818	0.081	27.82	0.735	0.126	26.71	0.737	0.108	29.99	0.849	0.087	24.47	0.873	0.095
D-Flow	26.44	0.573	0.180	28.60	0.746	0.167	25.20	0.616	0.188	32.79	0.898	0.044	27.02	0.840	0.081
Flow-Priors	29.67	0.756	0.165	27.28	0.726	0.188	23.95	0.566	0.274	32.95	0.909	0.051	26.13	0.809	0.130
PnP-Flow1	31.62	0.866	0.139	28.71	0.779	0.285	27.56	0.780	0.158	33.71	0.923	0.038	26.45	0.895	0.111
PnP-Flow5	31.87	0.868	0.170	29.01	0.785	0.312	28.01	0.791	0.167	34.44	0.932	0.045	27.14	0.900	0.127
Flower1-OT	32.12	0.881	0.113	29.38	0.793	0.237	26.76	0.759	0.262	33.68	0.922	0.042	26.67	0.914	0.065
Flower5-OT	32.75	0.892	0.126	29.73	0.801	0.264	27.09	0.767	0.272	34.36	0.931	0.048	27.32	0.922	0.067
Ours ($N = 2$)	33.31	0.904	0.081	29.89	0.809	0.229	29.10	0.816	0.167	34.71	0.936	0.043	29.27	0.924	0.060
Ours ($N = 4$)	32.93	0.898	0.079	29.42	0.796	0.188	28.50	0.801	0.141	34.30	0.932	0.039	28.65	0.921	0.052
Ours ($N = 10$)	31.96	0.879	0.072	28.72	0.775	0.145	27.75	0.777	0.115	33.53	0.922	0.033	27.94	0.916	0.047
Ours ($N = 25$)	29.69	0.776	0.081	28.16	0.754	0.119	26.54	0.676	0.121	32.91	0.912	0.030	26.76	0.819	0.077

Table 3: Computation time, memory usage, and number of function evaluations (NFEs) for the deblurring task on CelebA (128×128). Lower is better for all metrics. **Best** and **second** best results are shown in blue and red .

	Baselines							Ours			
	OT-ODE	D-Flow	Flow Priors	PnP-Flow1	PnP-Flow5	Flower1	Flower5	N=2	N=4	N=10	N=25
Time (s) ↓	6.37	299.25	41.27	2.33	11.46	2.67	13.03	0.24	0.50	1.21	3.13
Peak mem (MB) ↓	743	6124	2728	192	192	193	193	331	331	331	331
NFEs ↓	180	–	100	100	500	100	500	10	20	50	125

produces posterior-mean-like reconstructions, while finer integration moves toward perceptually realistic samples. Thus, N provides test-time control over the distortion–perception trade-off using a single trained model.

Iteration depth K. Table 4 evaluates $K \in \{3, 5, 7\}$ on random inpainting with N=2. Increasing K improves PSNR, while LPIPS is minimized at K=5. Since inference cost scales as $O(NK)$, we use K=5 throughout, balancing reconstruction quality with approximately 70% of the cost of K=7.

Robustness to noise-level and operator mismatch. Training with a fixed forward model ties the learned field to the operator $\boldsymbol{A}$ and noise level σ. We evaluate robustness to this mismatch in Tables 6 and 5. Table 6 applies a model trained with an inpainting mask ratio of 0.7 to ratios ranging from 0.5 to 0.8. At test time, the data-consistency step uses the actual test mask, while the learned regularizer and step-size parameters remain fixed at their p=0.7 values. Performance changes smoothly across this range and improves as fewer pixels are masked, indicating graceful transfer rather than failure. Because we do not train a matched model at each ratio, this experiment measures robustness but does not quantify the mismatch penalty. Table 5 evaluates a model trained at σ=0.01 on other noise levels. Despite this mismatch, our model outperforms the operator-agnostic Flower5-OT on all three metrics at both tested noise levels.

4.4 Additional Results

Appendix E provides additional quantitative and qualitative evaluations. Tables 11 and 12 report the results on larger test sets of 400 AFHQ-Cat and 1000 CelebA images, including standard deviations. The appendix also examines the distortion–perception trade-off (Figures 7 and 8), reconstruction trajectories (Figures 11-13), error maps (Figure 9), inference cost (Figure 10), operator mismatch (Figure 6), internal solver iterates (Figure 16), and posterior sample diversity (Figure 14). Appendix E.4 extends the evaluation to subsampled Fourier measurements, while Appendix E.3 studies posterior sampling and conditional-mean estimation on a toy problem.

Table 4: Iteration depth K

K	PSNR ↑	SSIM ↑	LPIPS ↓	ms ↓
3	34.63	0.959	0.020	137
5	35.03	0.963	0.018	223
7	35.21	0.964	0.019	322

Table 5: Noise-level mismatch.

Method	σ_{test}=0.005			σ_{test}=0.02		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Flower5-OT	34.02	0.954	0.019	33.65	0.947	0.024
Ours	35.19	0.965	0.018	34.57	0.954	0.020

5 Conclusion

We introduced FIRM, a measurement-conditional flow-matching method that incorporates explicit data-consistency updates into each evaluation of the learned velocity field. We characterized the measurement-conditioned posterior mean as the minimizer of a variational objective and proved that the velocity defined by this minimizer transports the source distribution to the posterior. Across five restoration tasks and two datasets, the method achieves leading distortion metrics at a fraction of the inference cost of competing flow-based solvers. Varying the number of sampling steps also controls the distortion-perception trade-off without retraining.

Table 6: Sensitivity to operator mismatch in random inpainting. A model trained at mask ratio p=0.7 (†) is evaluated at different test-time mask ratios without retraining. Although reconstruction quality decreases as more pixels are removed, performance does not exhibit abrupt degradation under mismatch.

p	PSNR ↑	SSIM ↑	LPIPS ↓
0.5	38.26	0.978	0.008
0.6	36.95	0.973	0.012
0.7†	35.03	0.963	0.018
0.8	32.10	0.936	0.034

Limitations. Our theoretical analysis assumes a known linear forward model with additive Gaussian noise. Extending the framework to nonlinear measurement processes and more general noise models is an important direction for future work. We also focus on linear interpolation and explicit Euler sampling; investigating alternative probability paths and numerical solvers may further improve sampling efficiency. Finally, our experiments concentrate on two-dimensional image restoration, and applications to other imaging modalities remain to be explored.

Reproducibility Statement

The method is specified completely by Algorithms 1, 2, and 3. Appendix B provide details of the algorithm. Appendix D lists the network architecture, optimizer settings, and all solver hyperparameters, with values collected in Table 7. Proofs are included in Appendix A. Datasets, degradation settings, and the evaluation protocol follow Martin et al. (2025) and Pourya et al. (2026) without modification, and are described in Section 4 and Appendix D. Source code is included.

AI use statement

Large language models were used to edit and polish the writing of this paper and to assist with routine code and LaTeX formatting.

References

Jonas Adler and Ozan Öktem. Learned Primal-Dual Reconstruction. *IEEE Trans. Med. Img.*, 37(6): 1322–1332, 2018.

Hemant K. Aggarwal, Merry P. Mani, and Mathews Jacob. MoDL: Model-Based Deep Learning Architecture for Inverse Problems. *IEEE Trans. Med. Img.*, 38(2):394–405, 2019.

Shadab Ahamed and Eldad Haber. DAWN-FM: Data-aware and Noise-informed Flow Matching for Solving Inverse Problems. *arXiv:2412.04766*, 2024.

Rizwan Ahmad, Charles A Bouman, Gregery T Buzzard, Stanley Chan, Sizhuo Liu, Edward T Reehorst, and Philip Schniter. Plug-and-play Methods for Magnetic Resonance Imaging: Using Denoisers for Image Recovery. *IEEE Signal Process Mag.*, 37(1):105–116, 2020.

Michael Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In *Proc. ICLR*, 2023.

Michael Samuel Albergo, Mark Goldstein, Nicholas Matthew Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic Interpolants with Data-Dependent Couplings. In *Proc. ICML*, 2024.

Amir Beck and Marc Teboulle. A fast Iterative Shrinkage-thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through Flows for Controlled Generation. In *Proc. ICML*, 2024.

Mario Bertero, Patrizia Boccacci, and Christine De Mol. *Introduction to Inverse Problems in Imaging*. CRC press, 2021.

Yochai Blau and Tomer Michaeli. The Perception-Distortion Tradeoff. In *Proc. CVPR*, pp. 6228–6237, 2018.

Jannis Chemseddine, Paul Hagemann, Gabriele Steidl, and Christian Wald. Conditional Wasserstein Distances with Applications in Bayesian OT Flow Matching. *J. Mach. Learn. Res.*, 26(141):1–47, 2025.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Proc. CVPR*, pp. 8188–8197, 2020.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *Proc. ICLR*, 2023a. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Hyungjin Chung, Jeongsol Lee, and Jong Chul Ye. Direct Diffusion Bridge Using Data Consistency for Inverse Problems. In *Proc. NeurIPS*, 2023b.

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A Survey on Diffusion Models for Inverse Problems. *arXiv:2410.00083*, 2024.

Mauricio Delbracio and Peyman Milanfar. Inversion by Direct Iteration: An Alternative to Denoising Diffusion for Image Restoration. *TMLR*, 2023.

Noam Elata, Hyungjin Chung, Jong Chul Ye, Tomer Michaeli, and Michael Elad. InvFussion: Bridging Supervised and Zero-Shot Diffusion for Inverse Problems. In *Proc. NeurIPS*, 2025.

Mário AT Figueiredo and Robert D Nowak. An EM Algorithm for Wavelet-based Image Restoration. *IEEE Trans. Image Proc.*, 12(8):906–916, 2003.

Dror Freirich, Tomer Michaeli, and Ron Meir. A Theory of the Distortion-Perception Tradeoff in Wasserstein Space. In *Proc. NeurIPS*, volume 34, pp. 25661–25672, 2021.

Karol Gregor and Yann LeCun. Learning Fast Approximations of Sparse Coding. In *Proc. ICML*, pp. 399–406, 2010.

Rémi Gribonval and Pierre Machart. Reconciling "Priors" and "Priors" without Prejudice? *Proc. NeurIPS*, 2013.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Proc. NeurIPS*, 2020.

Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient Step Denoiser for Convergent Plug-and-Play. In *Proc. ICLR*, 2022.

Ulugbek S. Kamilov, Charles B. Bouman, Gregery T. Buzzard, and Brendt Wohlberg. Plug-and-play methods for integrating physical and learned models in computational imaging. *IEEE Signal Process. Mag.*, 40(1):85–97, January 2023.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising Diffusion Restoration Models. In *Proc. NeurIPS*, 2022.

Jeongsol Kim, Bryan Kim, Jaemin Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. In *Proc. ICCV*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. *Proc. ICLR*, 2023.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2SB: Image-to-Image Schrödinger Bridge. In *Proc. ICML*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *Proc. ICLR*, 2023b.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proc. ICCV*, pp. 3730–3738, 2015.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A Variational Perspective on Solving Inverse Problems with Diffusion Models. In *Proc. ICLR*, 2024.

Ségolène Tiffany Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. PnP-Flow: Plug-and-Play Image Restoration with Flow Matching. In *Proc. ICLR*, 2025.

Michael T McCann, Kyong Hwan Jin, and Michael Unser. Convolutional Neural Networks for Inverse Problems in Imaging: A Review. *IEEE Signal Process Mag.*, 34(6):85–95, 2017.

Ali Mohammad-Djafari. Regularization, Bayesian inference, and Machine Learning Methods for Inverse Problems. *Entropy*, 23(12):1673, 2021.

Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing. *IEEE Signal Process Mag.*, 38(2):18–44, 2021.

Guy Ohayon, Tomer Michaeli, and Michael Elad. Posterior-Mean Rectified Flow: Towards Minimum MSE Photo-Realistic Image Restoration. In *Proc. ICLR*, 2025.

Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep Learning Techniques for Inverse Problems in Imaging. *IEEE J. Sel. Areas Inf. Theory*, 1(1):39–56, 2020.

Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free Linear Image Inverses via Flows. *TMLR*, 2024.

Mehrsa Pourya, Bassam El Rawas, and Michael Unser. Flower: A Flow-Matching Solver for Inverse Problems. In *Proc. ICLR*, 2026.

Haina Qin, Wenyang Luo, Libin Wang, Dandan Zheng, Jingdong Chen, Ming Yang, Bing Li, and Weiming Hu. Reversing Flow for Image Restoration. In *Proc. CVPR*, 2025.

Yaniv Romano, Michael Elad, and Peyman Milanfar. The Little Engine that Could: Regularization by Denoising (RED). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-Play Methods Provably Converge with Properly Trained Denoisers. In *Proc. ICML*, 2019.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-Image Diffusion Models. In *ACM SIGGRAPH Conf. Proc.*, pp. 1–10, 2022a.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image Super-resolution via Iterative Refinement. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4713–4726, 2022b.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proc. ICLR*, 2021.

Anuroop Sriram, Jure Zbontar, Tullie Murrell, Aaron Defazio, C. Lawrence Zitnick, Nafissa Yakubova, Florian Knoll, and Patricia Johnson. End-to-End Variational Networks for Accelerated MRI Reconstruction. In *Proc. MICCAI*, pp. 64–73, 2020.

Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

Matthieu Terris, Samuel Hurault, Maxime Song, and Julián Tachella. Reconstruct Anything Model: A Lightweight Foundation Model for Computational Imaging. In *Proc. ICLR*, 2026.

Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C., 1977.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and Generalizing Flow-based Generative Models with Minibatch Optimal Transport. *TMLR*, pp. 1–34, 2024.

Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play Priors for Model Based Reconstruction. In *IEEE Glob. Conf. Signal Inf. Process.*, pp. 945–948, 2013.

Curtis R Vogel. *Computational methods for inverse problems*. SIAM, 2002.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *Proc. ICLR*, 2023.

Jian Xu, Delu Zeng, John Paisley, and Qibin Zhao. What Do Flow-Based Inverse Solvers Approximate? A Posterior-Transport View. *arXiv preprint arXiv:2606.24516*, 2026.

Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for Compressive Sensing MRI. In *Proc. NeurIPS*, 2016.

Zongsheng Yue, Jianyi Wang, and Chen Change Loy. ResShift: Efficient Diffusion Model for Image Super-Resolution by Residual Shifting. In *Proc. NeurIPS*, 2023.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing. In *Proc. CVPR*, 2025.

Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proc. CVPR*, pp. 1828–1837, 2018.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-Play Image Restoration with Deep Denoiser Prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10): 6360–6376, 2021.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proc. CVPR*, pp. 586–595, 2018.

Yasi Zhang, Peiyu Yu, Yaxuan Zhu, Yingshan Chang, Feng Gao, Ying Nian Wu, and Oscar Leong. Flow Priors for Linear Inverse Problems via Iterative Corrupted Trajectory Matching. In *Proc. NeurIPS*, 2024.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising Diffusion Models for Plug-and-Play Image Restoration. In *Proc. CVPR*, 2023.

Nicolas Zilberstein, Morteza Mardani, and Santiago Segarra. Flow Map Denoisers: Traversing the Distortion-Perception Plane for Inverse Problems. *arXiv:2606.19802*, 2026.

A THEORETICAL RESULTS PROOFS

Proof roadmap. The proof follows the same two-stage construction as the method. First, we reduce the joint observation $(\boldsymbol{x}_t, \boldsymbol{y})$ to an equivalent Gaussian denoising problem. The Gaussian MMSE characterization of Gribonval & Machart (2013) then gives the variational form of the posterior mean. Second, we show that this mean determines the conditional velocity, that the velocity satisfies the conditional continuity equation, and that its ODE transports the Gaussian source toward the posterior. The lemmas below verify these steps separately, and the final proof combines them.

A.1 VARIATIONAL CHARACTERIZATION

The Gaussian MMSE characterization connects posterior-mean estimation to variational optimization. For a signal observed in Gaussian noise, the posterior mean is the unique minimizer of a quadratic data-fitting term plus an implicit regularizer determined by the signal distribution and noise covariance. We will use this result to express the measurement-conditioned posterior mean as the solution of an optimization problem.

Lemma 1 (Gaussian MMSE characterization (Gribonval & Machart, 2013, Cor. 1))**.** *Let $\boldsymbol{z}$ have a non-degenerate distribution on $\mathbb{R}^n$, and let*

$$\boldsymbol{u} = \boldsymbol{z} + \boldsymbol{\xi}, \qquad \boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}), \qquad \boldsymbol{\Sigma} \succ 0, \tag{28}$$

where $\boldsymbol{\xi}$ is independent of $\boldsymbol{z}$. Then there exists a function $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that

$$\mathbb{E}[\boldsymbol{z} \mid \boldsymbol{u}] = \operatorname*{arg\,min}_{\boldsymbol{z}' \in \mathbb{R}^n} \left\{ \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{z}'\|^2_{\boldsymbol{\Sigma}^{-1}} + \phi(\boldsymbol{z}') \right\}. \tag{29}$$

The minimizer is the unique global minimizer and stationary point, and ϕ is unique up to an additive constant.

Proof. This is Corollary 1 of Gribonval & Machart (2013), restated using $\|\boldsymbol{a}\|^2_{\mathbf{B}} = \boldsymbol{a}^\top \mathbf{B} \boldsymbol{a}$. We use only the existence and uniqueness conclusions in Eq. (29). □

The interpolation state $\boldsymbol{x}_t$ and measurement $\boldsymbol{y}$ are both noisy linear observations of the clean image $\boldsymbol{x}_1$. Because their noise terms are independent, the two observations can be stacked into a single linear Gaussian model with block-diagonal noise covariance.

Lemma 2. *For $t \in (0, 1)$, define*

$$\widetilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{y} \end{bmatrix}, \qquad \widetilde{\boldsymbol{A}} = \begin{bmatrix} t\boldsymbol{I} \\ \boldsymbol{A} \end{bmatrix}, \qquad \widetilde{\boldsymbol{\Sigma}} = \begin{bmatrix} (1-t)^2 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma^2 \boldsymbol{I} \end{bmatrix}. \tag{30}$$

Then

$$\widetilde{\boldsymbol{y}} = \widetilde{\boldsymbol{A}} \boldsymbol{x}_1 + \widetilde{\boldsymbol{b}}, \qquad \widetilde{\boldsymbol{b}} \sim \mathcal{N}(\boldsymbol{0}, \widetilde{\boldsymbol{\Sigma}}), \tag{31}$$

and $\widetilde{\boldsymbol{b}}$ is independent of $\boldsymbol{x}_1$.

Proof. The interpolation equation gives $\boldsymbol{x}_t = t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0$, and the measurement equation gives $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{\eta}$. Therefore,

$$\widetilde{\boldsymbol{b}} = \begin{bmatrix} (1-t)\boldsymbol{x}_0 \\ \boldsymbol{\eta} \end{bmatrix}. \tag{32}$$

Because $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ are independent of each other and of $\boldsymbol{x}_1$, the stated covariance and independence follow. □

The augmented model is linear Gaussian, but it is not yet in the additive denoising form required by Lemma 1. We combine the interpolation state and measurement according to their noise levels to form $\boldsymbol{u}_t$. The resulting variable is equivalent to observing $\boldsymbol{x}_1$ directly under additive Gaussian noise, with precision matrix $\boldsymbol{M}_t$.

Lemma 3. *Let*

$$\boldsymbol{M}_t = \widetilde{\boldsymbol{A}}^\top \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{A}} = \frac{t^2}{(1-t)^2} \boldsymbol{I} + \frac{1}{\sigma^2} \boldsymbol{A}^\top \boldsymbol{A} \tag{33}$$

and define

$$\boldsymbol{u}_t = \boldsymbol{M}_t^{-1} \widetilde{\boldsymbol{A}}^\top \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{y}}. \tag{34}$$

Then $\boldsymbol{M}_t \succ 0$ *and*

$$\boldsymbol{u}_t = \boldsymbol{x}_1 + \boldsymbol{\xi}_t, \qquad \boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{M}_t^{-1}), \tag{35}$$

where $\boldsymbol{\xi}_t$ *is independent of* $\boldsymbol{x}_1$.

Proof. For every nonzero $\boldsymbol{a} \in \mathbb{R}^n$,

$$\boldsymbol{a}^\top \boldsymbol{M}_t \boldsymbol{a} = \frac{t^2}{(1-t)^2} \|\boldsymbol{a}\|_2^2 + \frac{1}{\sigma^2} \|\boldsymbol{A}\boldsymbol{a}\|_2^2 > 0. \tag{36}$$

Hence $\boldsymbol{M}_t \succ 0$. Substituting Eq. (31) into Eq. (34) gives

$$\boldsymbol{u}_t = \boldsymbol{x}_1 + \boldsymbol{M}_t^{-1} \widetilde{\boldsymbol{A}}^\top \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{b}} =: \boldsymbol{x}_1 + \boldsymbol{\xi}_t. \tag{37}$$

The noise $\boldsymbol{\xi}_t$ is Gaussian and independent of $\boldsymbol{x}_1$. Its covariance is

$$\mathrm{Cov}(\boldsymbol{\xi}_t) = \boldsymbol{M}_t^{-1} \widetilde{\boldsymbol{A}}^\top \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{A}} \boldsymbol{M}_t^{-1} = \boldsymbol{M}_t^{-1}. \tag{38}$$

This proves Eq. (35). □

The reduced observation $\boldsymbol{u}_t$ is useful only if it preserves all the information about $\boldsymbol{x}_1$ contained in $(\boldsymbol{x}_t, \boldsymbol{y})$. The following lemma shows that the augmented data-fitting term equals a Gaussian denoising term involving $\boldsymbol{u}_t$, up to a constant independent of the unknown image. Consequently, conditioning on $\boldsymbol{u}_t$ gives the same posterior mean as conditioning on the original observations.

Lemma 4. *The statistic* $\boldsymbol{u}_t$ *in Eq. (34) is sufficient for* $\boldsymbol{x}_1$ *in* $\widetilde{\boldsymbol{y}}$. *Consequently,*

$$\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t, \boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}_1 \mid \widetilde{\boldsymbol{y}}] = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{u}_t]. \tag{39}$$

Moreover, for every $\boldsymbol{x} \in \mathbb{R}^n$,

$$\frac{1}{2(1-t)^2} \|\boldsymbol{x}_t - t\boldsymbol{x}\|_2^2 + \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 = \frac{1}{2} \|\boldsymbol{u}_t - \boldsymbol{x}\|_{\boldsymbol{M}_t}^2 + c(\widetilde{\boldsymbol{y}}),$$

where $c(\widetilde{\boldsymbol{y}})$ *does not depend on* $\boldsymbol{x}$.

Proof. By the definitions of $\boldsymbol{M}_t$ and $\boldsymbol{u}_t$,

$$\widetilde{\boldsymbol{A}}^\top \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{y}} = \boldsymbol{M}_t \boldsymbol{u}_t. \tag{40}$$

Expanding the augmented quadratic and completing the square therefore gives

$$\frac{1}{2} \|\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{A}}\boldsymbol{x}\|_{\widetilde{\boldsymbol{\Sigma}}^{-1}}^2 = \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{M}_t \boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{M}_t \boldsymbol{u}_t + \frac{1}{2} \widetilde{\boldsymbol{y}}^\top \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{y}} = \frac{1}{2} \|\boldsymbol{u}_t - \boldsymbol{x}\|_{\boldsymbol{M}_t}^2 + c(\widetilde{\boldsymbol{y}}),$$

where

$$c(\widetilde{\boldsymbol{y}}) = \frac{1}{2} \widetilde{\boldsymbol{y}}^\top \widetilde{\boldsymbol{\Sigma}}^{-1} \widetilde{\boldsymbol{y}} - \frac{1}{2} \boldsymbol{u}_t^\top \boldsymbol{M}_t \boldsymbol{u}_t. \tag{41}$$

Expanding the left-hand side blockwise yields Eq. (40).

The conditional likelihood consequently factors as

$$p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x}_1) = r(\widetilde{\boldsymbol{y}}) \exp\left(-\frac{1}{2} \|\boldsymbol{u}_t - \boldsymbol{x}_1\|_{\boldsymbol{M}_t}^2 \right), \tag{42}$$

where $r(\widetilde{\boldsymbol{y}})$ is independent of $\boldsymbol{x}_1$. The factorization criterion shows that $\boldsymbol{u}_t$ is sufficient for $\boldsymbol{x}_1$, which proves Eq. (39). □

The preceding lemmas reduce the observations $(\boldsymbol{x}_t, \boldsymbol{y})$ to an equivalent Gaussian denoising observation $\boldsymbol{u}_t$ without losing information about $\boldsymbol{x}_1$. We can therefore apply the Gaussian MMSE characterization to $\boldsymbol{u}_t$ and then rewrite its quadratic term using the original observations. The resulting objective contains one term associated with the interpolation state and another that explicitly enforces the measurement model.

Lemma 5. *Suppose that p_1 is non-degenerate. For every $t \in (0,1)$, there exists a function $\phi_t : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, unique up to an additive constant, such that*

$$\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t, \boldsymbol{y}] = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \frac{1}{2(1-t)^2} \|\boldsymbol{x}_t - t\boldsymbol{x}\|_2^2 + \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \phi_t(\boldsymbol{x}) \right\}. \tag{43}$$

The posterior mean is the unique global minimizer and stationary point. The dependence of ϕ_t on the fixed operator $\boldsymbol{A}$ and noise level σ is suppressed for readability.

Proof. Lemma 3 expresses $\boldsymbol{u}_t$ as the Gaussian denoising model

$$\boldsymbol{u}_t = \boldsymbol{x}_1 + \boldsymbol{\xi}_t, \qquad \boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{M}_t^{-1}). \tag{44}$$

Applying Lemma 1 with noise covariance $\boldsymbol{M}_t^{-1}$ gives a function ϕ_t for which

$$\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{u}_t] = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \frac{1}{2} \|\boldsymbol{u}_t - \boldsymbol{x}\|_{\boldsymbol{M}_t}^2 + \phi_t(\boldsymbol{x}) \right\}. \tag{45}$$

Lemma 4 shows that the conditional mean equals $\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t, \boldsymbol{y}]$ and that the quadratic term differs from the two data terms in Eq. (43) only by a constant independent of $\boldsymbol{x}$. Adding or removing that constant does not change the minimizer or stationary points, proving the result. □

A.2 Conditional probability flow

The variational result characterizes the posterior mean, whereas flow matching requires a velocity field. Under linear interpolation, the velocity of each trajectory can be written using only its endpoint $\boldsymbol{x}_1$ and current state $\boldsymbol{x}_t$. Averaging over the possible endpoints conditioned on $\boldsymbol{x}_t$ and $\boldsymbol{y}$ therefore expresses the optimal conditional velocity directly in terms of the posterior mean.

Lemma 6. *For $t \in [0,1)$, the optimal measurement-conditional velocity is*

$$\boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{y}) = \mathbb{E}[\boldsymbol{x}_1 - \boldsymbol{x}_0 \mid \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}] = \frac{\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}] - \boldsymbol{x}}{1-t}. \tag{46}$$

Proof. The linear path has velocity $\dot{\boldsymbol{x}}_t = \boldsymbol{x}_1 - \boldsymbol{x}_0$. Since

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - t\boldsymbol{x}_1}{1-t}, \tag{47}$$

we obtain

$$\boldsymbol{x}_1 - \boldsymbol{x}_0 = \frac{\boldsymbol{x}_1 - \boldsymbol{x}_t}{1-t}. \tag{48}$$

Taking the conditional expectation given $\boldsymbol{x}_t = \boldsymbol{x}$ and $\boldsymbol{y}$ proves Eq. (46). □

To establish a probability flow, we must first show that the posterior mean changes smoothly with the current state $\boldsymbol{x}$. The bounded-support assumption allows differentiation through the conditional density. The resulting Jacobian has a simple covariance form, which will be used to control the regularity of the conditional velocity.

Lemma 7 (Regularity of the conditional mean)**.** *Suppose* $\operatorname{supp} p_1 \subseteq \overline{B}(\boldsymbol{0}, R)$ *and fix $\boldsymbol{y}$. Define*

$$\boldsymbol{m}_t(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}]. \tag{49}$$

For every $t \in [0,1)$, the conditional density $p_t(\boldsymbol{x} \mid \boldsymbol{y})$ is positive and smooth, and $\boldsymbol{m}_t$ is smooth in $(\boldsymbol{x}, t)$. Moreover, $\|\boldsymbol{m}_t(\boldsymbol{x})\|_2 \le R$ and

$$\nabla_{\boldsymbol{x}} \boldsymbol{m}_t(\boldsymbol{x}) = \frac{t}{(1-t)^2} \operatorname{Cov}(\boldsymbol{x}_1 \mid \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}). \tag{50}$$

Proof. Let $\nu = p(\boldsymbol{x}_1 \mid \boldsymbol{y})$. Because the Gaussian likelihood is positive, ν is supported on $\overline{B}(\mathbf{0}, R)$. For $t < 1$, the conditional density of $\boldsymbol{x}_t$ is

$$p_t(\boldsymbol{x} \mid \boldsymbol{y}) = \int \varphi_{1-t}(\boldsymbol{x} - t\boldsymbol{x}_1)\, \nu(\mathrm{d}\boldsymbol{x}_1), \tag{51}$$

where φ_s denotes the density of $\mathcal{N}(\mathbf{0}, s^2\boldsymbol{I})$. The Gaussian kernel is positive and smooth. Because ν has bounded support, its derivatives are dominated uniformly on compact subsets of $\mathbb{R}^n \times [0, 1)$. We may therefore differentiate under the integral sign.

The conditional distribution is supported on $\overline{B}(\mathbf{0}, R)$, so its mean satisfies $\|\boldsymbol{m}_t(\boldsymbol{x})\|_2 \le R$.

Bayes' rule gives

$$q_{\boldsymbol{x},t}(\mathrm{d}\boldsymbol{x}_1) = \frac{\varphi_{1-t}(\boldsymbol{x} - t\boldsymbol{x}_1)}{p_t(\boldsymbol{x} \mid \boldsymbol{y})}\nu(\mathrm{d}\boldsymbol{x}_1). \tag{52}$$

Differentiating its log-density with respect to $\boldsymbol{x}$ yields

$$\nabla_{\boldsymbol{x}} \log q_{\boldsymbol{x},t}(\boldsymbol{x}_1) = \frac{t}{(1-t)^2}\big(\boldsymbol{x}_1 - \boldsymbol{m}_t(\boldsymbol{x})\big). \tag{53}$$

Since $\boldsymbol{m}_t(\boldsymbol{x}) = \int \boldsymbol{x}_1 q_{\boldsymbol{x},t}(\mathrm{d}\boldsymbol{x}_1)$, differentiation under the integral gives Eq. (50). □

A unique probability flow requires the conditional velocity to vary regularly with the current state. The Jacobian identity from Lemma 7 allows us to bound this variation through the conditional covariance of $\boldsymbol{x}_1$. Bounded support provides a uniform covariance bound, making the velocity globally Lipschitz on every time interval $[0, \tau]$ with $\tau < 1$.

Lemma 8 (Lipschitz conditional velocity)**.** *Under the assumptions of Lemma 7, $\boldsymbol{v}(\cdot, t, \boldsymbol{y})$ is globally Lipschitz for every $t \in [0, 1)$, with*

$$\mathrm{Lip}(\boldsymbol{v}(\cdot, t, \boldsymbol{y})) \le \frac{1}{1-t} + \frac{tR^2}{(1-t)^3}. \tag{54}$$

The bound is uniform over $t \in [0, \tau]$ for every $\tau < 1$.

Proof. Using Lemma 6 and Lemma 7,

$$\nabla_{\boldsymbol{x}} \boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{y}) = \frac{t}{(1-t)^3}\,\mathrm{Cov}(\boldsymbol{x}_1 \mid \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}) - \frac{1}{1-t}\boldsymbol{I}. \tag{55}$$

For every unit vector $\boldsymbol{a}$,

$$\boldsymbol{a}^\top \mathrm{Cov}(\boldsymbol{x}_1 \mid \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y})\boldsymbol{a} \le \mathbb{E}[(\boldsymbol{a}^\top \boldsymbol{x}_1)^2 \mid \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}] \le R^2. \tag{56}$$

Thus the conditional covariance has operator norm at most R^2, giving Eq. (54). For every $\tau < 1$,

$$\sup_{t \in [0,\tau]} \mathrm{Lip}(\boldsymbol{v}(\cdot, t, \boldsymbol{y})) \le \frac{1}{1-\tau} + \frac{\tau R^2}{(1-\tau)^3} < \infty. \tag{57}$$

At $t = 0$, the identity $\boldsymbol{v}(\boldsymbol{x}, 0, \boldsymbol{y}) = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{y}] - \boldsymbol{x}$ gives Lipschitz constant one, consistent with the bound. □

For each possible endpoint $\boldsymbol{x}_1$, linear interpolation defines a simple Gaussian probability path with a known velocity. The conditional distribution $p_t(\boldsymbol{x} \mid \boldsymbol{y})$ is obtained by averaging these paths over the posterior distribution of $\boldsymbol{x}_1$. Averaging their probability fluxes yields the optimal measurement-conditional velocity and establishes the corresponding continuity equation.

Lemma 9 (Conditional continuity equation)**.** *Under the assumptions of Lemma 8, the conditional density $p_t(\boldsymbol{x} \mid \boldsymbol{y})$ satisfies*

$$\partial_t p_t(\boldsymbol{x} \mid \boldsymbol{y}) + \nabla \cdot (p_t(\boldsymbol{x} \mid \boldsymbol{y})\boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{y})) = 0, \qquad t \in [0, 1). \tag{58}$$

Proof. Fix $\boldsymbol{y}$ and write $\nu = p(\boldsymbol{x}_1 \mid \boldsymbol{y})$. For a fixed endpoint $\boldsymbol{x}_1$, the density

$$p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = \varphi_{1-t}(\boldsymbol{x} - t\boldsymbol{x}_1) \tag{59}$$

is induced by the deterministic map $\boldsymbol{x}_0 \mapsto t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0$. It therefore satisfies

$$\partial_t p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) + \nabla \cdot \left(p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \frac{\boldsymbol{x}_1 - \boldsymbol{x}}{1-t} \right) = 0. \tag{60}$$

Because $\boldsymbol{x}_0$ is independent of $(\boldsymbol{x}_1, \boldsymbol{y})$,

$$p_t(\boldsymbol{x} \mid \boldsymbol{y}) = \int p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)\, \nu(\mathrm{d}\boldsymbol{x}_1). \tag{61}$$

Integrating Eq. (60) with respect to ν gives

$$\partial_t p_t(\boldsymbol{x} \mid \boldsymbol{y}) = -\nabla \cdot \int p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \frac{\boldsymbol{x}_1 - \boldsymbol{x}}{1-t}\, \nu(\mathrm{d}\boldsymbol{x}_1). \tag{62}$$

By Bayes' rule and Lemma 6, the integral in Eq. (62) equals

$$p_t(\boldsymbol{x} \mid \boldsymbol{y}) \mathbb{E}\left[\frac{\boldsymbol{x}_1 - \boldsymbol{x}}{1-t} \,\middle|\, \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y} \right] = p_t(\boldsymbol{x} \mid \boldsymbol{y}) \boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{y}). \tag{63}$$

Substitution proves Eq. (58). □

The continuity equation identifies the probability path associated with the conditional velocity, but a well-defined sampler also requires the corresponding ODE to have unique trajectories. The Lipschitz bound guarantees a unique flow on every interval that ends before $t = 1$. Because the ODE flow and the conditional path start from the same Gaussian distribution and satisfy the same continuity equation, their distributions agree. The conditional path then converges to the measurement-conditioned posterior as t approaches one.

Lemma 10 (Conditional transport and posterior limit)**.** *Under the assumptions of Lemma 8, for every $\tau < 1$ the ODE*

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{x}_t, t, \boldsymbol{y}) \tag{64}$$

generates a unique flow map $\Phi_{0\to\tau}$ satisfying

$$(\Phi_{0\to\tau})_{\#}\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) = p_\tau(\cdot \mid \boldsymbol{y}). \tag{65}$$

Furthermore,

$$p_\tau(\cdot \mid \boldsymbol{y}) \Rightarrow p(\boldsymbol{x}_1 \mid \boldsymbol{y}) \qquad \text{as } \tau \to 1^-. \tag{66}$$

Proof. Lemma 7 and Lemma 8 show that the velocity is continuous in time and globally Lipschitz in $\boldsymbol{x}$, uniformly on every interval $[0, \tau]$ with $\tau < 1$. It also has at most linear growth in $\boldsymbol{x}$. The Picard–Lindelöf theorem therefore gives a unique global flow map on each such interval.

The pushforward of the initial distribution through this flow satisfies the continuity equation associated with $\boldsymbol{v}$. By Lemma 9, $p_t(\cdot \mid \boldsymbol{y})$ satisfies the same equation. The two solutions also have the same initial distribution because $\boldsymbol{x}_0$ is independent of $\boldsymbol{y}$:

$$p_0(\cdot \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{67}$$

Uniqueness of the transported measure under a globally Lipschitz velocity proves Eq. (65).

Finally, couple the conditional path by drawing $\boldsymbol{x}_1 \sim p(\boldsymbol{x}_1 \mid \boldsymbol{y})$ and $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ independently. Then

$$\boldsymbol{x}_\tau = \tau\boldsymbol{x}_1 + (1-\tau)\boldsymbol{x}_0 \longrightarrow \boldsymbol{x}_1 \qquad \text{almost surely.} \tag{68}$$

Almost-sure convergence implies convergence in distribution, proving Eq. (66). The result requires no extension of the flow map to $t = 1$. □

A.3 Proof of Proposition 1

Proof. Lemma 5 proves that the measurement-conditioned posterior mean is the unique minimizer of Eq. (13). Lemma 6 converts this mean into the optimal conditional velocity. Finally, Lemmas 9 and 10 show that the resulting velocity defines the conditional probability flow and converges weakly to the measurement-conditioned posterior. These are the two claims of Proposition 1. □

B Algorithm Details

This section provides implementation details of the HQS iterations used to parameterize the conditional velocity. The main text gives the HQS objective and its two alternating subproblems in Eq. (17) and Eq. (18).

B.1 Data-Consistency Update

The $\boldsymbol{x}$-subproblem is

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \frac{\mu}{2} \|\boldsymbol{x} - \boldsymbol{z}^k\|_2^2 \right\}. \tag{69}$$

Its optimality condition is

$$\left(\sigma^{-2}\boldsymbol{A}^\top \boldsymbol{A} + \mu \boldsymbol{I}\right) \boldsymbol{x}^{k+1} = \sigma^{-2}\boldsymbol{A}^\top \boldsymbol{y} + \mu \boldsymbol{z}^k. \tag{70}$$

Defining $\rho = \mu\sigma^2$ gives the normal equations used by the implementation,

$$\left(\boldsymbol{A}^\top \boldsymbol{A} + \rho \boldsymbol{I}\right) \boldsymbol{x}^{k+1} = \boldsymbol{A}^\top \boldsymbol{y} + \rho \boldsymbol{z}^k. \tag{71}$$

The update depends explicitly on the forward operator $\boldsymbol{A}$. Closed-form solutions for the operators considered in this work are given in Appendix B.4.

B.2 Learned Regularization Update

The exact $\boldsymbol{z}$-subproblem is

$$\boldsymbol{z}_\star^{k+1} = \arg\min_{\boldsymbol{z}} \left\{ \phi_t(\boldsymbol{z}) + \frac{1}{2(1-t)^2} \|\boldsymbol{x}_t - t\boldsymbol{z}\|_2^2 + \frac{\mu}{2} \|\boldsymbol{x}^{k+1} - \boldsymbol{z}\|_2^2 \right\}. \tag{72}$$

The two quadratic terms can be written as the weighted residual

$$\frac{1}{2} \left\| \widetilde{\boldsymbol{y}}_k - \widetilde{\boldsymbol{A}}_k \boldsymbol{z} \right\|_{\widetilde{\boldsymbol{\Sigma}}_k^{-1}}^2, \tag{73}$$

where

$$\widetilde{\boldsymbol{y}}_k = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}^{k+1} \end{bmatrix}, \qquad \widetilde{\boldsymbol{A}}_k = \begin{bmatrix} t\boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix}, \qquad \widetilde{\boldsymbol{\Sigma}}_k = \begin{bmatrix} (1-t)^2 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \mu^{-1}\boldsymbol{I} \end{bmatrix}. \tag{74}$$

Completing the square reduces the two observations to the sufficient statistic

$$\boldsymbol{u}_k = \frac{\dfrac{t}{(1-t)^2}\boldsymbol{x}_t + \mu \boldsymbol{x}^{k+1}}{\dfrac{t^2}{(1-t)^2} + \mu}, \tag{75}$$

with precision

$$\boldsymbol{M}_k = \left(\frac{t^2}{(1-t)^2} + \mu \right) \boldsymbol{I}. \tag{76}$$

Consequently, Eq. (72) is equivalent, up to terms independent of $\boldsymbol{z}$, to

$$\boldsymbol{z}_\star^{k+1} = \arg\min_{\boldsymbol{z}} \left\{ \phi_t(\boldsymbol{z}) + \frac{1}{2} \|\boldsymbol{u}_k - \boldsymbol{z}\|_{\boldsymbol{M}_k}^2 \right\}. \tag{77}$$

Thus, the exact update denoises a precision-weighted combination of the interpolation state and the data-consistent iterate. The reduction identifies the form of the regularization subproblem but does not imply that an arbitrary minimizer of Eq. (77) is a posterior mean. In particular, the converse direction of the MMSE characterization in Gribonval & Machart (2013) does not hold in general.

Because ϕ_t is unavailable, we follow PnP practice (Venkatakrishnan et al., 2013) and replace the exact update with a learned estimator,

$$\widehat{\boldsymbol{z}}^{k+1} = \mathsf{R}_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{k+1}, \boldsymbol{x}_t, t \right). \tag{78}$$

Algorithm 3 Operator-aware posterior-mean estimation

Require: Interpolation state $\boldsymbol{x}_t$, time $t < 1$, measurement $\boldsymbol{y}$, forward operator $\boldsymbol{A}$, learned estimator $\mathsf{R}_{\boldsymbol{\theta}}$, iteration count K, coupling ρ, and damping β
Ensure: Posterior-mean estimate $\boldsymbol{z}^K$
1: $\boldsymbol{z}^0 \leftarrow \boldsymbol{x}_t$
2: **for** $k = 0, \dots, K-1$ **do**
3: $\quad \boldsymbol{x}^{k+1} \leftarrow \left(\boldsymbol{A}^\top \boldsymbol{A} + \rho \boldsymbol{I}\right)^{-1} \left(\boldsymbol{A}^\top \boldsymbol{y} + \rho \boldsymbol{z}^k\right)$
4: $\quad \widehat{\boldsymbol{z}}^{k+1} \leftarrow \mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{x}^{k+1}, \boldsymbol{x}_t, t)$
5: $\quad \boldsymbol{z}^{k+1} \leftarrow (1-\beta)\boldsymbol{x}^{k+1} + \beta \widehat{\boldsymbol{z}}^{k+1}$
6: **return** $\boldsymbol{z}^K$

The network receives the two components of Eq. (75) separately, allowing their relative weighting to depend on t and the HQS parameters.

Following Ryu et al. (2019), we damp the learned estimate toward the data-consistent iterate,

$$\boldsymbol{z}^{k+1} = (1-\beta)\boldsymbol{x}^{k+1} + \beta \widehat{\boldsymbol{z}}^{k+1}, \qquad \beta \in [0, 1]. \tag{79}$$

Damping retains an explicit contribution from $\boldsymbol{x}^{k+1}$ and prevents the learned estimator from completely overwriting the data-consistent iterate.

B.3 Velocity Parameterization and Cost

Starting from $\boldsymbol{z}^0 = \boldsymbol{x}_t$, we alternate Eq. (71), Eq. (78), and Eq. (79) for K iterations. The final iterate estimates the measurement-conditioned posterior mean,

$$\boldsymbol{z}_{\boldsymbol{\theta}}^K(\boldsymbol{x}_t, t, \boldsymbol{y}) \approx \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t, \boldsymbol{y}], \tag{80}$$

and defines the learned conditional velocity through Eq. (19).

The network parameters $\boldsymbol{\theta}$ enter only through $\mathsf{R}_{\boldsymbol{\theta}}$. Each velocity evaluation requires K data-consistency solves and K learned-estimator evaluations. Since every data-consistency solve depends on $\boldsymbol{A}$ and $\boldsymbol{y}$, the forward operator acts within every inner iteration of the learned velocity field.

Algorithm 3 summarizes the operator-aware posterior mean evaluation.

B.4 Operator-Specific Data-Consistency Solutions

Image denoising. For denoising, $\boldsymbol{A} = \boldsymbol{I}$, and Eq. (71) gives

$$\boldsymbol{x}^{k+1} = \frac{\boldsymbol{y} + \rho \boldsymbol{z}^k}{1 + \rho}. \tag{81}$$

The update is a weighted average of the measurement and the current iterate.

Image inpainting. For inpainting, $\boldsymbol{A} = \boldsymbol{M}$ is a diagonal binary mask satisfying $\boldsymbol{M}^\top \boldsymbol{M} = \boldsymbol{M}$. Equation Eq. (71) can therefore be solved elementwise,

$$x_i^{k+1} = \begin{cases} \dfrac{y_i + \rho z_i^k}{1 + \rho}, & i \in \Omega, \\ z_i^k, & i \notin \Omega, \end{cases} \tag{82}$$

where Ω denotes the observed pixels.

Image super-resolution. We implement super-resolution using stride subsampling without an anti-aliasing prefilter. The forward operator retains a periodic subset of pixels, and $\boldsymbol{A}^\top \boldsymbol{A}$ is a diagonal binary mask. The update is therefore Eq. (82), with $\boldsymbol{y}$ replaced by $\boldsymbol{A}^\top \boldsymbol{y}$ and Ω denoting the sampled grid locations.

Image deblurring. For deblurring, $\boldsymbol{A} = \mathbf{H}$ is a spatially invariant blur. Under circular boundary conditions, $\mathbf{H} = \boldsymbol{F}^H \boldsymbol{\Lambda} \boldsymbol{F}$, where $\boldsymbol{\Lambda}$ contains the Fourier coefficients of the blur kernel. Equation Eq. (71) then gives

$$\boldsymbol{x}^{k+1} = \boldsymbol{F}^H \left[\frac{\overline{\boldsymbol{\Lambda}}\, \boldsymbol{F}\boldsymbol{y} + \rho \boldsymbol{F}\boldsymbol{z}^k}{|\boldsymbol{\Lambda}|^2 + \rho} \right], \tag{83}$$

where all operations inside the brackets are elementwise.

Subsampled Fourier measurements. For the compressed-sensing experiments in Appendix E.4, let $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{F}$, where $\boldsymbol{F}$ is the orthonormal two-dimensional DFT and $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{p})$ is the sampling mask. Writing $\widehat{\boldsymbol{z}}^{,k} = \boldsymbol{F}\boldsymbol{z}^k$, the Fourier-domain update is

$$\widehat{x}_i^{,k+1} = \begin{cases} \dfrac{y_i + \rho \widehat{z}_i^{,k}}{1+\rho}, & i \in \Omega, \\ \widehat{z}_i^{,k}, & i \notin \Omega, \end{cases} \qquad \boldsymbol{x}^{k+1} = \mathrm{Re}\left\{ \boldsymbol{F}^H \widehat{\boldsymbol{x}}^{,k+1} \right\}. \tag{84}$$

For the conjugate-symmetric mask used in our experiments, taking the real part is equivalent to solving the corresponding real normal equations. The operator is applied independently to each color channel.

General linear operators. For operators without exploitable structure, Eq. (71) can be solved using conjugate gradients. All operators used in this work admit the closed-form updates described above.

C Related Works

Generative solvers for imaging inverse problems differ mainly in whether the forward operator $\boldsymbol{A}$ enters only at inference, through an operator-aware architecture, or during conditional training. We organize prior work along this axis.

Inference-time guidance with a pretrained prior. Diffusion-based methods combine an unconditional prior with measurement consistency at sampling time. DPS and ΠGDM approximate likelihood guidance using denoised clean-image estimates (Chung et al., 2023a; Song et al., 2023); DDRM and DDNM exploit the spectral or null-space structure of $\boldsymbol{A}$ (Kawar et al., 2022; Wang et al., 2023); and DiffPIR, variational PnP diffusion, and DAPS alternate generative updates with optimization, projection, or conditional resampling (Zhu et al., 2023; Mardani et al., 2024; Zhang et al., 2025). Flow-based counterparts include OT-ODE, D-Flow, Flow Priors, FlowDPS, PnP-Flow, and Flower (Pokle et al., 2024; Ben-Hamu et al., 2024; Zhang et al., 2024; Kim et al., 2025; Martin et al., 2025; Pourya et al., 2026). These methods can reuse a pretrained prior across many compatible operators, but construct only an approximation to the measurement-conditional dynamics at inference. A recent posterior-transport analysis places several such approximations in a common framework and bounds their posterior bias (Xu et al., 2026).

Amortized conditional models. Conditional restoration methods instead learn the measurement-to-image dynamics from paired data. Examples include SR3, Palette, InDI, I^2SB, conditional diffusion bridges, ResShift, and ResFlow (Saharia et al., 2022b;a; Delbracio & Milanfar, 2023; Liu et al., 2023a; Chung et al., 2023b; Yue et al., 2023; Qin et al., 2025). Within flow matching, data-dependent stochastic interpolants and conditional Wasserstein formulations provide theoretical foundations (Albergo et al., 2024; Chemseddine et al., 2025), while DAWN-FM conditions its velocity field on a measurement embedding, typically $\boldsymbol{A}^\top \boldsymbol{y}$, and the noise level (Ahamed & Haber, 2024). Most such models are trained for a particular task or degradation family and encode the measurement through $\boldsymbol{y}$, $\boldsymbol{A}^\top \boldsymbol{y}$, or a learned representation. Our distinction is that $\boldsymbol{A}$ acts explicitly through data-consistency updates inside the generative dynamics.

Operator-aware learned solvers. Plug-and-play methods retain the forward operator in a data-fidelity update while replacing the regularizer with a learned denoiser (Venkatakrishnan et al., 2013; Zhang et al., 2021; Hurault et al., 2022). Algorithm unrolling makes these iterations end-to-end

trainable, yielding methods such as ISTA-Net, ADMM-Net, learned primal–dual, MoDL, and VarNet (Zhang & Ghanem, 2018; Yang et al., 2016; Adler & Öktem, 2018; Aggarwal et al., 2019; Sriram et al., 2020). MoDL is particularly close in its alternation between data consistency and a shared-weight learned update. These architectures are typically trained as point estimators and do not by themselves define posterior samplers. Our method imports their operator-aware structure into a generative velocity field by applying data consistency at every inner iteration of every ODE step.

Distortion–perception control. The distortion–perception trade-off is fundamental in image restoration (Blau & Michaeli, 2018). For MSE distortion and Wasserstein perception, optimal estimators trace a Wasserstein geodesic between the minimum-MSE and perfect-perception endpoints (Freirich et al., 2021). PMRF operationalizes this idea by transporting a posterior-mean estimate toward the data distribution (Ohayon et al., 2025); related interpolation mechanisms appear in InDI and data-dependent interpolants (Delbracio & Milanfar, 2023; Albergo et al., 2024). Flow-map denoisers similarly produce a one-parameter family of estimators that traverses the distortion–perception plane, with optimality established in the Gaussian setting (Zilberstein et al., 2026). Our Euler step count N plays an analogous empirical role. Under the ideal conditional velocity, one Euler step reduces to a conditional-mean update; for a learned velocity field, the $N{=}1$ output should be interpreted as posterior-mean-like rather than exactly equal to the posterior mean. Increasing N evaluates the learned estimator at progressively lower interpolation noise levels, where posterior modes become more resolvable.

Our model is trained for a specified family of forward operators and noise levels, placing it in the amortized conditional family. The central question is whether, given operator-specific training, explicitly embedding $\boldsymbol{A}$ in the dynamics improves upon conditioning a network only on an image-space measurement. We investigate this question in Appendix E.2. We also compare with inference-time flow solvers because they are established reference points for the benchmark (Pokle et al., 2024; Ben-Hamu et al., 2024; Zhang et al., 2024; Martin et al., 2025; Pourya et al., 2026). Methods that amortize across families of operators (Terris et al., 2026; Elata et al., 2025) provide a natural direction for extending our approach.

D Implementation Details

Estimator network. The learned estimator $\mathsf{R}_{\boldsymbol{\theta}}$ is a U-Net that takes the concatenation of the data-consistent iterate $\boldsymbol{x}^{k+1}$ and the flow state $\boldsymbol{x}_t$ along the channel dimension ($2C$ input channels) and is conditioned on t through a sinusoidal embedding of width $4\,\mathrm{ch}$ injected into every residual block. Blocks use GroupNorm with 32 groups and Swish activations, with self-attention at the resolutions listed in Table 7. Weights are shared across the K inner iterations, so the parameter count of the full model equals that of a single U-Net. CelebA models use $\mathrm{ch} = 32$ with multipliers $(1, 2, 4, 8)$ (34.5M parameters); AFHQ-Cat models use multipliers $(1, 2, 4, 8, 8)$ with $\mathrm{ch} = 32$ (59.9M), except deblurring which uses $\mathrm{ch} = 64$ (239.4M).

Solver. We use $K = 5$ solver inner iterations, damping $\beta = 0.5$, and a fixed coupling $\rho = 0.01$ for every operator and noise level. Writing the data-consistency step as $\boldsymbol{x}^{k+1} = (\boldsymbol{A}^\top \boldsymbol{A} + \rho \boldsymbol{I})^{-1}(\boldsymbol{A}^\top \boldsymbol{y} + \rho \boldsymbol{z}^k)$ absorbs the noise variance into $\rho = \mu\sigma^2$; the solve is closed-form for all operators we consider (Appendix B). The iteration is initialized at $\boldsymbol{z}^0 = \boldsymbol{x}_t$.

Training. AdamW with learning rate 10^{-4} decayed by a cosine schedule to 10^{-6}, weight decay 10^{-4}, gradient-norm clipping at 1.0, and no mixed precision. We train with a total batch size of 32 (8 per GPU on 4 GPUs, DDP) for 500 epochs of 2000 steps on CelebA and 700 epochs of 750 steps on AFHQ-Cat, where each epoch resamples the training set with replacement. An exponential moving average of the weights with decay 0.999 is maintained from epoch 20 and used for all validation and test-time evaluation. For the first 20 epochs we train the estimator without data-consistency, using a single network call on the concatenation of $\boldsymbol{A}^\top \boldsymbol{y}$ and $\boldsymbol{x}_t$ and then enable the $K = 5$ data-term solver for the remainder of training; this warm start avoids backpropagating through five iterations of an untrained estimator. Times t are drawn by low-discrepancy stratified sampling over 64 equal strata of $[t_{\mathsf{min}}, t_{\max}]$ rather than i.i.d. uniformly, which reduces the variance of the epoch-averaged loss at small batch size. Training roughly takes between 24–48 hours.

Table 7: Hyperparameters. All values are held fixed across the five tasks unless noted.

		CelebA (128^2)	AFHQ-Cat (256^2)
Solver	iteration depth K	5	5
	damping β	0.5	0.5
	coupling ρ	0.01	0.01
	initialization $\boldsymbol{z}^0$	$\boldsymbol{x}_t$	$\boldsymbol{x}_t$
Flow	t_{min} / $t_{\max}$	10^{-3} / 0.995	10^{-3} / 0.995
	clamp τ_{min}	0.1	0.1
	integrator	Euler	Euler
Network	base width ch	32	32 (64 for deblurring)
	channel multipliers	$(1, 2, 4, 8)$	$(1, 2, 4, 8, 8)$
	residual blocks per level	6	6
	attention resolutions	16, 8	32, 16
	parameters	34.5 M	59.9 M (239.4 M)
Optimization	optimizer	AdamW	AdamW
	learning rate (cosine to 10^{-6})	10^{-4}	10^{-4}
	weight decay / grad clip	10^{-4} / 1.0	10^{-4} / 1.0
	batch size (total)	32	32
	epochs $\times$ steps per epoch	500×2000	700×750
	EMA decay (from epoch 20)	0.999	0.999

Evaluation protocol. PSNR is computed on $[0, 1]$ with `data_range`$= 1$, matching the postprocessing used by Pourya et al. (2026) so that the numbers are directly comparable; SSIM is computed on the clamped $[0, 1]$ image and LPIPS with an AlexNet backbone on the clamped $[-1, 1]$ image. All runs use seed 42.

D.1 Baselines

The flow baselines share the pretrained prior of Martin et al. (2025): 34.5M parameters on CelebA and 31.0M on AFHQ-Cat (computed from the released checkpoints). Our CelebA models match this capacity exactly; on AFHQ-Cat we use a larger estimator for deblurring, the hardest of our settings at 256×256, where smaller estimators underfit. Its parameters are shared across the K inner iterations.

PnP-HQS (DPIR). PnP-HQS is DPIR (Zhang et al., 2021), which is a half-quadratic splitting with a DRUNet denoiser prior, alternating the closed-form data step $(\boldsymbol{A}^\top \boldsymbol{A} + \rho_k \boldsymbol{I})\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{y} + \rho_k \boldsymbol{z}$ with a denoiser step at noise level σ_k, log-spaced from $49/255$ down to the measurement noise, with $\rho_k = \lambda\sigma^2/\sigma_k^2$ and $\lambda = 0.23$ (DPIR default). Operators and measurements are taken from our own implementation, so the degradations are identical to those used by our method and the other baselines. To avoid handicapping the baseline with a generic prior, we train the DRUNet (32.6M parameters) on the same data and preprocessing as our own models, AFHQ-Cat at 256×256, the CelebA denoiser is trained analogously at 128×128. Iteration counts are tuned per task to the plateau of a validation subset. On AFHQ-Cat, 24 for deblurring and super-resolution, 100 for random inpainting, which needs many more iterations to propagate the fill across missing pixels, and 1 for denoising, where $\boldsymbol{A} = \boldsymbol{I}$ makes the data step trivial and DPIR reduces to a single denoiser pass (more iterations are worse, 32.32 and 32.36 dB at 8 and 16 iterations, against 32.69 dB for the single pass). Same number of iterations was used for CelebA experiments as well. Super-resolution is initialized from a bicubic upsampling of the measurement, as in DPIR; the remaining tasks use $\boldsymbol{x}_0 = \boldsymbol{A}^\top \boldsymbol{y}$. We do not report box inpainting, since a Gaussian denoiser is non-generative and cannot fill a large contiguous hole.

ISTA-Net$^+$. We use ISTA-Net$^+$ (Zhang & Ghanem, 2018) as a representative deep-unrolling baseline. We use K proximal-gradient steps, each a data-fidelity gradient step $\boldsymbol{x} \leftarrow \boldsymbol{x} - \lambda_k \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$ followed by a learned proximal (convolutional forward transform, soft-threshold, inverse transform, residual), where we trained end-to-end with the standard reconstruction plus symmetry loss ($\gamma = 0.01$). We adapt it to three-channel images and to the measurement operators used in this

Table 8: Damping β

β	PSNR ↑	SSIM ↑	LPIPS ↓
0.25	35.30	0.965	0.018
0.50	35.03	0.963	0.018
0.75	35.17	0.964	0.018
1.00	34.70	0.961	0.020

Table 9: Ablation of the operator embedding on CelebA (100 images). *Flow-only* is the same U-Net trained under the same objective on the concatenation of $\boldsymbol{A}^\top \boldsymbol{y}$ and $\boldsymbol{x}_t$, without the HQS solver, costing one network call per Euler step against our K=5; rows are matched by step count N, so our entries use five times the function evaluations. Better of each pair in bold.

		Random inpainting			Deblurring		
N	Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
2	Flow-only	34.54	0.955	**0.0186**	35.77	0.952	**0.0281**
	Ours	**35.03**	**0.963**	0.0189	**36.04**	**0.957**	0.0295
4	Flow-only	34.15	0.953	0.0172	35.45	0.950	**0.0256**
	Ours	**34.67**	**0.962**	**0.0169**	**35.78**	**0.956**	0.0280
10	Flow-only	33.53	0.948	0.0149	34.72	0.943	**0.0209**
	Ours	**33.98**	**0.956**	**0.0146**	**35.03**	**0.950**	0.0235
25	Flow-only	33.07	0.943	0.0136	**34.03**	**0.934**	**0.0174**
	Ours	**33.42**	**0.951**	**0.0134**	31.46	0.848	0.0265

paper. We use $K = 9$ layers with 32 channels and unshared weights (0.35M parameters), and train one model per task with Adam, learning rate 10^{-4}, initialized at $\boldsymbol{A}^\top \boldsymbol{y}$ (bicubic for super-resolution). Box inpainting and denoising are omitted since compact convolutional proximal cannot fill a large contiguous hole.

E Additional Results

E.1 Additional Ablation

Damping coefficient β. Table 8 reports quality for $\beta \in \{0.25, 0.5, 0.75, 1.0\}$ on random inpainting at N=2, K=5. The coefficient sets how much of each update comes from the learned estimator rather than the data-consistent iterate, so β=1 discards the data-consistent iterate from the update (though $\boldsymbol{x}^{k+1}$ still enters R_θ as an input) and is worst on all three metrics. Among the damped settings differences are small, all ahead of every baseline in Table 1. Thus we use β=0.5 for all tasks rather than tuning per setting.

E.2 Ablating the HQS

We evaluate whether embedding the measurement operator $\boldsymbol{A}$ in the generative dynamics improves upon conditioning a network on the measurement. Here, we compare this effect by using the same network, objective, training schedule, but we concatenate $\boldsymbol{A}^\top \boldsymbol{y}$ as an additional conditioning channel with $\boldsymbol{x}_t$ as input in place of the HQS solver. The cost will be reduced, since we will not have the inner iterations of HQS. We report the results for random inpainting and deblurring tasks, across different ODE step-sizes in Table 9.

At equal compute, our model at N=2 (10 NFEs) improves over Flow-only at N=10 (10 NFEs) by 1.5 dB PSNR on random inpainting and 1.3 dB on deblurring, with higher SSIM, while Flow-only attains lower LPIPS. This is consistent with the distortion–perception trade-off controlled by N: more Euler steps move either model toward a more perceptual operating point. At matched step count, our model improves PSNR and SSIM in all but one setting.

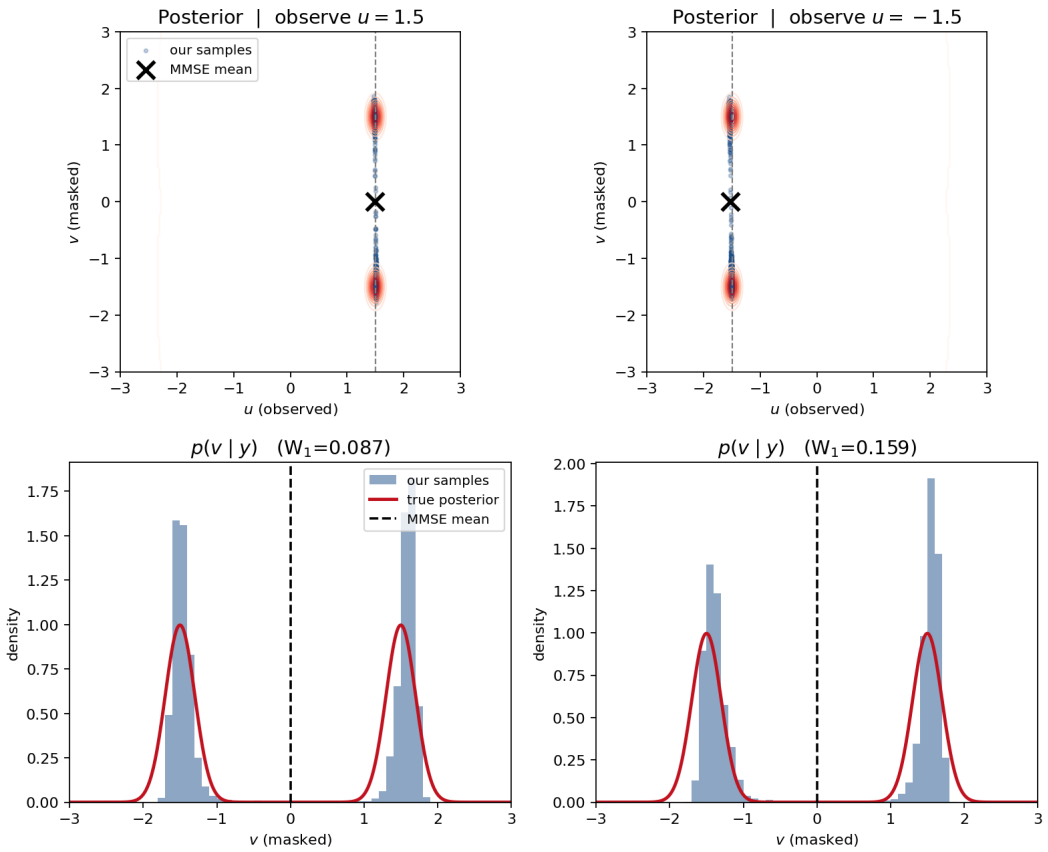

Figure 4: Posterior recovery at N=50. **Top:** samples (blue) and analytical posterior contours (red) for measurements near $u = \pm 1.5$; the posterior mean ($\times$) lies between the modes. **Bottom:** sampled and analytical marginals $p(v|\boldsymbol{y})$.

E.3 Posterior recovery on a toy problem

To evaluate posterior fidelity against ground truth, we consider a two-dimensional problem with an analytical posterior. Let $\boldsymbol{x} = (u, v) \in \mathbb{R}^2$ follow an equally weighted four-component Gaussian mixture with means $(\pm 1.5, \pm 1.5)$ and component standard deviation 0.2. We observe $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta}$ with $\boldsymbol{A} = \mathrm{diag}(1, 0)$ and $\sigma = 0.1$. Thus, u is observed while $p(v|\boldsymbol{y})$ remains bimodal, with its mean at the low-density point $v = 0$. We replace the U-Net with a small MLP, retain K=5, β=0.5, and ρ=0.01, and draw 4000 samples for each fixed measurement.

At N=50, the sampler recovers both posterior modes with empirical weights $0.49/0.51$, compared with the exact $0.50/0.50$ (Figure 4). The masked-coordinate distance is $W_1(v) \approx 0.1$, and the observed coordinate agrees with the measurement within ± 0.02. The remaining error is primarily due to mild under-dispersion of the sampled modes.

Effect of the number of sampling steps. Table 10 and Figure 5 show that the N=1 output collapses to the posterior-mean region, whereas increasing N progressively resolves both modes. Mode coverage reaches 77% at N=2 and 93% at N=4; for $6 \leq N \leq 25$, the recovered weights and mode locations are within 1% and 0.05 of their analytical values.

For uniform Euler integration, $\Delta t = 1/N = 1 - t_{N-1}$, so the final update satisfies

$$\boldsymbol{x}_N = \mathbb{E}_{\boldsymbol{\theta}}(\boldsymbol{x}_{N-1}, t_{N-1}, \boldsymbol{y}), \qquad t_{N-1} = \frac{N-1}{N}.$$

Table 10: Posterior recovery versus N for $u{=}+1.5$ using 4000 samples. The analytical posterior has negligible mass in $|v| < 0.75$, equal mode weights, and modes near ± 1.5.

N	1	2	4	50
$W_1(v) \downarrow$	1.44	0.44	0.13	0.11
mass in $\lvert v\rvert < 0.75$	100%	23%	7%	0.5%
mode weights	—	0.50/0.50	0.52/0.48	0.49/0.51
empirical modes	0	± 1.26	± 1.45	± 1.52

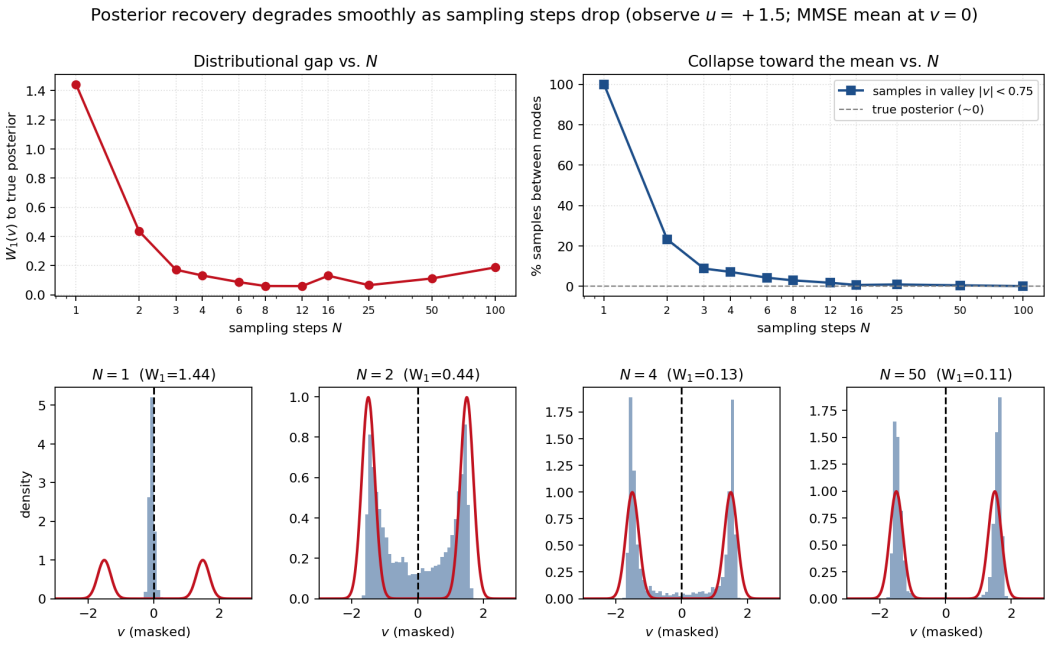

Figure 5: Posterior recovery versus N. **Top:** $W_1(v)$ and mass in the low-density valley. **Bottom:** sampled (blue) and analytical (red) marginals. The modes become resolved near $N \approx 6$, followed by mild drift and under-dispersion at large N.

Increasing N therefore moves the final estimator evaluation closer to $t = 1$, where the interpolation noise is smaller, while also changing the state produced by the preceding Euler steps. This explains the transition from posterior-mean-like estimates at small N to mode-resolving samples at larger N, consistent with the distortion–perception trade-off observed in the image experiments.

E.4 Compressed-sensing Fourier sampling.

We consider subsampled Fourier measurements $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{F}$, the forward model used in accelerated MRI, where $\boldsymbol{F}$ is the two-dimensional DFT and $\boldsymbol{P}$ a binary sampling mask. We evaluate a Cartesian mask retaining 21.88% of k-space on AFHQ-Cat, with complex Gaussian noise of standard deviation $\sigma = 0.002$, applying the operator channel by channel for color images. We compare against Flower (Pourya et al., 2026) with $\gamma = 0$, $N = 100$ and $N_{\text{avg}} = 1$, and against the zero-filled adjoint $\boldsymbol{A}^*\boldsymbol{y}$. Figure 17 and Table 13 report the results.

Table 11: Results on 1000 CelebA images. **Best** and **second** best results are shown in blue and red .

	Metric	Degraded	OT-ODE	PnP-Flow5		Flower5-OT		Ours			
				1	5	1	5	$N = 2$	$N = 4$	$N = 10$	$N = 25$
Denoising	PSNR↑	20.00 ± 0.03	30.48 ± 1.13	31.79 ± 1.21	32.27 ± 1.25	32.25 ± 1.19	33.10 ± 1.21	33.68 ± 1.18	33.25 ± 1.20	32.27 ± 1.23	31.58 ± 1.22
	SSIM↑	0.351 ± 0.058	0.856 ± 0.029	0.903 ± 0.028	0.909 ± 0.028	0.912 ± 0.024	0.924 ± 0.021	0.931 ± 0.019	0.927 ± 0.020	0.914 ± 0.023	0.901 ± 0.025
	LPIPS↓	0.367 ± 0.087	0.033 ± 0.014	0.044 ± 0.020	0.056 ± 0.024	0.033 ± 0.014	0.038 ± 0.016	0.034 ± 0.013	0.034 ± 0.013	0.031 ± 0.012	0.028 ± 0.010
Deblurring	PSNR↑	27.82 ± 1.20	33.00 ± 1.68	34.56 ± 1.72	34.89 ± 1.79	35.01 ± 1.72	35.71 ± 1.80	36.09 ± 1.72	35.82 ± 1.77	35.05 ± 1.80	31.46 ± 0.98
	SSIM↑	0.741 ± 0.028	0.919 ± 0.024	0.935 ± 0.016	0.940 ± 0.016	0.946 ± 0.016	0.953 ± 0.015	0.957 ± 0.015	0.955 ± 0.015	0.949 ± 0.017	0.848 ± 0.039
	LPIPS↓	0.123 ± 0.035	0.029 ± 0.014	0.037 ± 0.018	0.044 ± 0.020	0.025 ± 0.012	0.031 ± 0.014	0.029 ± 0.013	0.027 ± 0.011	0.023 ± 0.009	0.026 ± 0.013
Super-resolution	PSNR↑	10.38 ± 1.31	31.43 ± 3.50	31.12 ± 1.57	31.53 ± 1.58	32.41 ± 1.75	33.14 ± 1.80	34.52 ± 2.00	34.10 ± 2.06	33.27 ± 2.05	30.48 ± 1.29
	SSIM↑	0.185 ± 0.053	0.903 ± 0.075	0.900 ± 0.030	0.904 ± 0.031	0.921 ± 0.023	0.930 ± 0.022	0.949 ± 0.017	0.946 ± 0.019	0.937 ± 0.021	0.838 ± 0.040
	LPIPS↓	0.829 ± 0.063	0.026 ± 0.063	0.045 ± 0.017	0.054 ± 0.020	0.034 ± 0.013	0.039 ± 0.015	0.028 ± 0.012	0.027 ± 0.010	0.023 ± 0.008	0.028 ± 0.012
Random inpainting	PSNR↑	12.13 ± 1.51	28.87 ± 1.89	33.23 ± 2.24	34.20 ± 2.29	33.26 ± 2.25	34.15 ± 2.29	35.27 ± 2.51	34.93 ± 2.61	34.23 ± 2.64	33.66 ± 2.61
	SSIM↑	0.199 ± 0.057	0.872 ± 0.037	0.944 ± 0.019	0.954 ± 0.017	0.945 ± 0.019	0.953 ± 0.017	0.963 ± 0.016	0.962 ± 0.017	0.957 ± 0.019	0.951 ± 0.022
	LPIPS↓	1.027 ± 0.077	0.050 ± 0.023	0.017 ± 0.007	0.021 ± 0.008	0.017 ± 0.007	0.019 ± 0.007	0.018 ± 0.009	0.016 ± 0.007	0.014 ± 0.006	0.013 ± 0.006
Box inpainting	PSNR↑	22.29 ± 1.49	29.68 ± 2.56	30.44 ± 2.42	30.99 ± 2.49	31.19 ± 2.63	31.81 ± 2.69	33.28 ± 2.44	32.63 ± 2.50	31.95 ± 2.52	31.45 ± 2.48
	SSIM↑	0.745 ± 0.038	0.921 ± 0.021	0.934 ± 0.017	0.940 ± 0.016	0.946 ± 0.013	0.953 ± 0.012	0.961 ± 0.012	0.959 ± 0.013	0.955 ± 0.013	0.949 ± 0.014
	LPIPS↓	0.211 ± 0.035	0.035 ± 0.017	0.036 ± 0.018	0.042 ± 0.020	0.021 ± 0.014	0.022 ± 0.015	0.022 ± 0.012	0.021 ± 0.011	0.020 ± 0.010	0.019 ± 0.010

Table 12: Results on 400 AFHQ-Cat images. **Best** and **second** best results are shown in blue and red .

	Metric	Degraded	OT-ODE	PnP-Flow5		Flower5-OT		Ours			
				1	5	1	5	$N = 2$	$N = 4$	$N = 10$	$N = 25$
Denoising	PSNR↑	20.00 ± 0.01	30.07 ± 1.53	31.22 ± 1.60	31.47 ± 1.63	31.70 ± 1.60	32.35 ± 1.56	32.90 ± 1.55	32.50 ± 1.58	31.51 ± 1.63	29.39 ± 1.15
	SSIM↑	0.311 ± 0.072	0.813 ± 0.032	0.861 ± 0.034	0.862 ± 0.035	0.876 ± 0.029	0.888 ± 0.027	0.901 ± 0.022	0.894 ± 0.024	0.874 ± 0.028	0.776 ± 0.026
	LPIPS↓	0.511 ± 0.114	0.079 ± 0.021	0.142 ± 0.037	0.176 ± 0.044	0.113 ± 0.033	0.127 ± 0.036	0.082 ± 0.026	0.080 ± 0.024	0.072 ± 0.023	0.078 ± 0.022
Deblurring	PSNR↑	24.16 ± 1.62	27.23 ± 2.31	28.18 ± 2.41	28.48 ± 2.42	28.85 ± 2.45	29.20 ± 2.47	29.37 ± 2.49	28.88 ± 2.49	28.15 ± 2.47	27.57 ± 2.44
	SSIM↑	0.522 ± 0.046	0.713 ± 0.076	0.764 ± 0.070	0.769 ± 0.070	0.777 ± 0.068	0.786 ± 0.067	0.795 ± 0.064	0.782 ± 0.067	0.758 ± 0.070	0.735 ± 0.073
	LPIPS↓	0.437 ± 0.052	0.125 ± 0.035	0.302 ± 0.064	0.331 ± 0.067	0.253 ± 0.058	0.282 ± 0.063	0.244 ± 0.059	0.199 ± 0.051	0.151 ± 0.042	0.124 ± 0.035
Super-resolution	PSNR↑	11.88 ± 1.72	26.13 ± 2.54	27.15 ± 2.42	27.61 ± 2.43	26.41 ± 2.10	26.73 ± 2.10	28.56 ± 2.54	27.94 ± 2.58	27.19 ± 2.56	26.08 ± 2.16
	SSIM↑	0.219 ± 0.073	0.717 ± 0.077	0.765 ± 0.065	0.777 ± 0.062	0.743 ± 0.069	0.751 ± 0.067	0.803 ± 0.064	0.786 ± 0.069	0.760 ± 0.075	0.665 ± 0.058
	LPIPS↓	0.878 ± 0.062	0.110 ± 0.029	0.168 ± 0.043	0.178 ± 0.045	0.275 ± 0.051	0.286 ± 0.053	0.179 ± 0.042	0.150 ± 0.036	0.122 ± 0.030	0.122 ± 0.026
Random inpainting	PSNR↑	13.56 ± 1.84	29.52 ± 2.21	33.15 ± 2.86	33.88 ± 2.87	33.12 ± 2.84	33.82 ± 2.84	34.12 ± 2.96	33.70 ± 3.11	32.93 ± 3.16	32.32 ± 3.12
	SSIM↑	0.230 ± 0.070	0.840 ± 0.044	0.918 ± 0.030	0.929 ± 0.027	0.918 ± 0.030	0.927 ± 0.027	0.933 ± 0.027	0.929 ± 0.031	0.918 ± 0.036	0.907 ± 0.039
	LPIPS↓	1.065 ± 0.080	0.088 ± 0.021	0.040 ± 0.012	0.047 ± 0.014	0.043 ± 0.012	0.049 ± 0.013	0.047 ± 0.016	0.042 ± 0.014	0.036 ± 0.012	0.032 ± 0.012
Box inpainting	PSNR↑	21.82 ± 2.03	24.76 ± 2.73	26.57 ± 3.23	27.19 ± 3.26	26.79 ± 3.51	27.47 ± 3.46	28.94 ± 2.53	28.29 ± 2.47	27.60 ± 2.42	26.52 ± 1.99
	SSIM↑	0.743 ± 0.038	0.873 ± 0.021	0.895 ± 0.020	0.900 ± 0.020	0.915 ± 0.014	0.923 ± 0.014	0.925 ± 0.012	0.922 ± 0.012	0.916 ± 0.013	0.825 ± 0.022
	LPIPS↓	0.201 ± 0.040	0.091 ± 0.026	0.110 ± 0.027	0.127 ± 0.029	0.065 ± 0.020	0.067 ± 0.019	0.059 ± 0.015	0.051 ± 0.013	0.045 ± 0.012	0.071 ± 0.022

Table 13: Compressed-sensing Fourier sampling on AFHQ-Cat (avg 100 images).

	CS with Cartesian (21.88%)				
Method	Zero-filled	Flower1-OT	Ours ($N = 2$)	Ours ($N = 4$)	Ours ($N = 10$)
PSNR ↑	25.62	31.17	32.06	31.71	31.06
SSIM ↑	0.725	0.870	0.892	0.885	0.869
LPIPS ↓	0.343	0.062	0.074	0.064	0.054



Figure 6: Reconstructions under forward-model mismatch on random inpainting (CelebA, 128×128). A single model trained at a 0.7 mask ratio with $\sigma = 0.01$ is applied to test measurements generated with mask ratios from 0.5 to 0.8. Quality is monotone in the test mask ratio, with no failure mode at either extreme. Without matched-ratio models these numbers characterize graceful degradation rather than quantifying the mismatch penalty. Quantitative values are reported in Table 6.

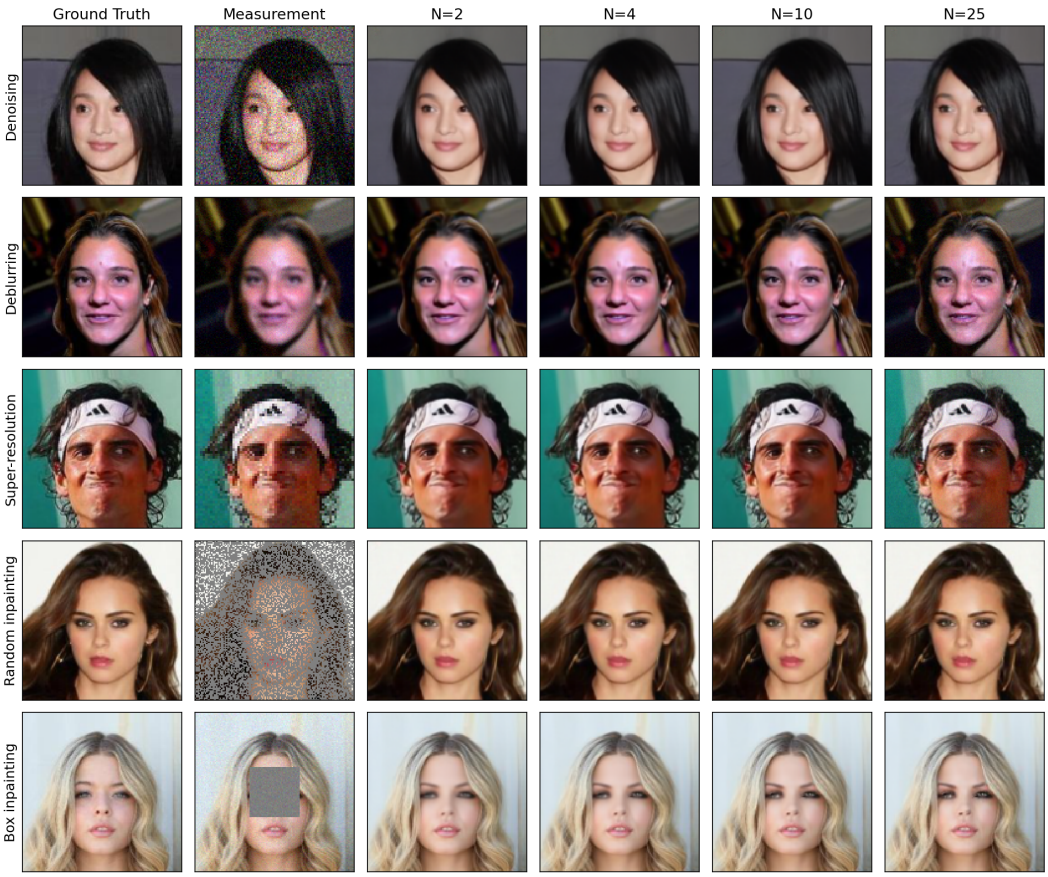

Figure 7: Distortion–perception trade-off on CelebA (128×128).



Figure 8: Distortion–perception trade-off on AFHQ (256×256).

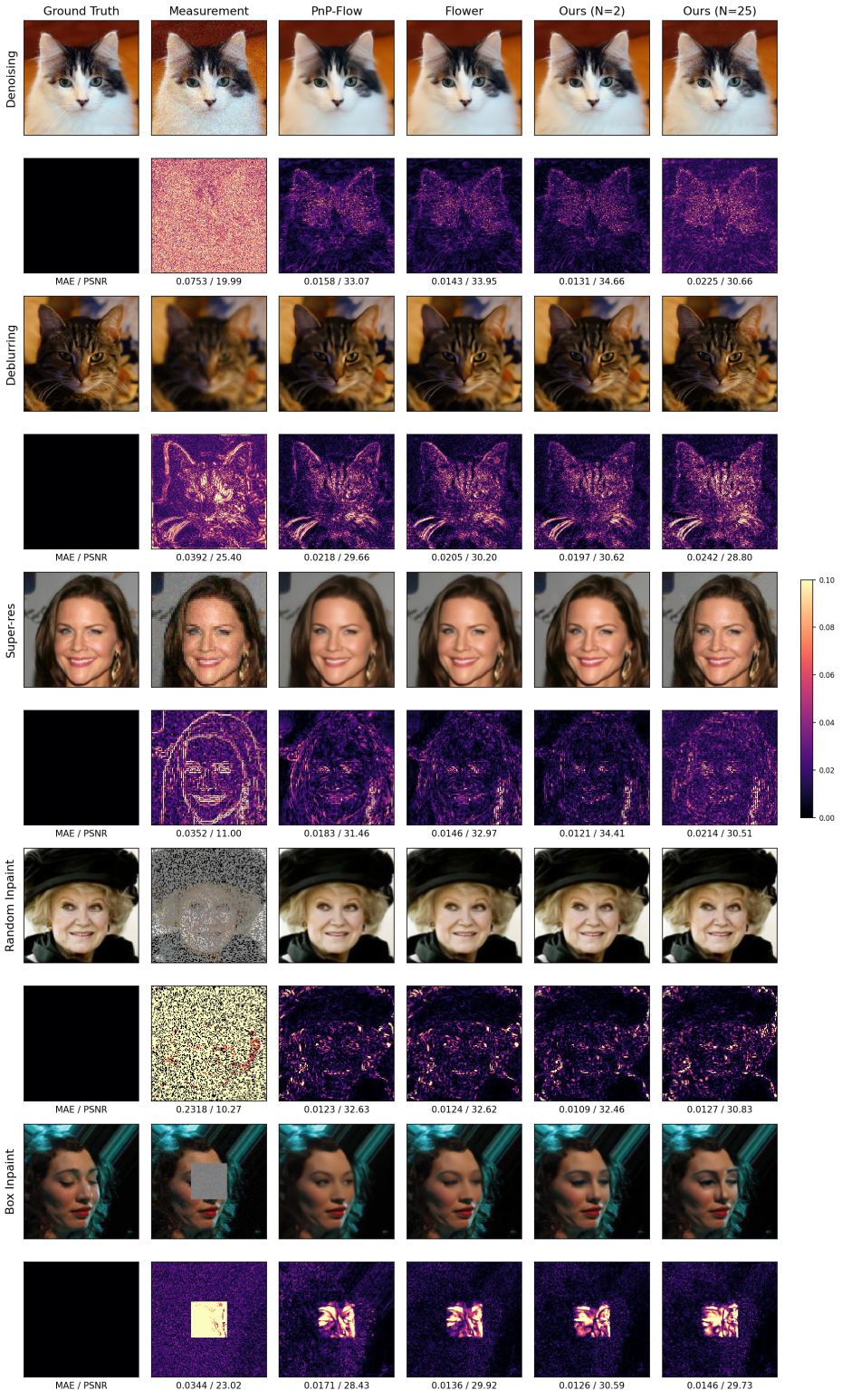

Figure 9: Reconstructions (top) and per-pixel error maps (bottom, magma, $[0, 0.1]$) for the five inverse problems (denoising, deblurring, ×2 super-resolution, random inpainting, box inpainting). Columns: ground truth, measurement, PnP-Flow, Flower, Ours (N=2), Ours (N=25). MAE / PSNR (dB) is reported under each panel. Ours (N=2) yields the lowest error on every task.

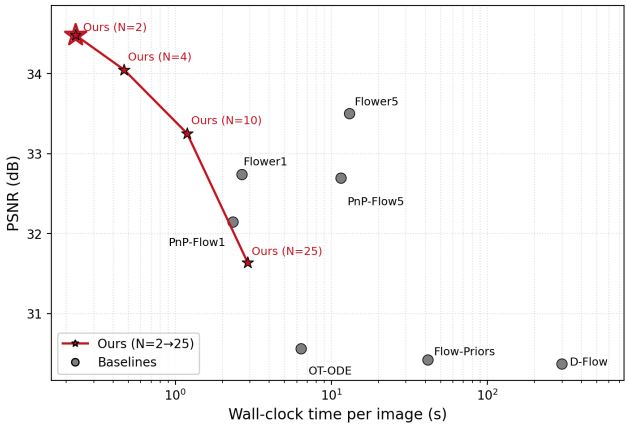

Figure 10: Reconstruction quality versus inference cost on CelebA (128×128). PSNR is averaged over the five restoration tasks of Table 1 (100 test images each); the horizontal axis is per-image wall-clock, measured on a single NVIDIA RTX A6000 and plotted on a log scale. Up and to the left is better. Our method is shown over Euler steps $N \in \{2, 4, 10, 25\}$ with $K{=}5$ fixed, while each baseline contributes a single point. At $N{=}2$ we exceed the strongest baseline (Flower5) at $\approx 50\times$ lower cost. PSNR decreases along our curve as N grows because additional steps trade distortion for perceptual quality.

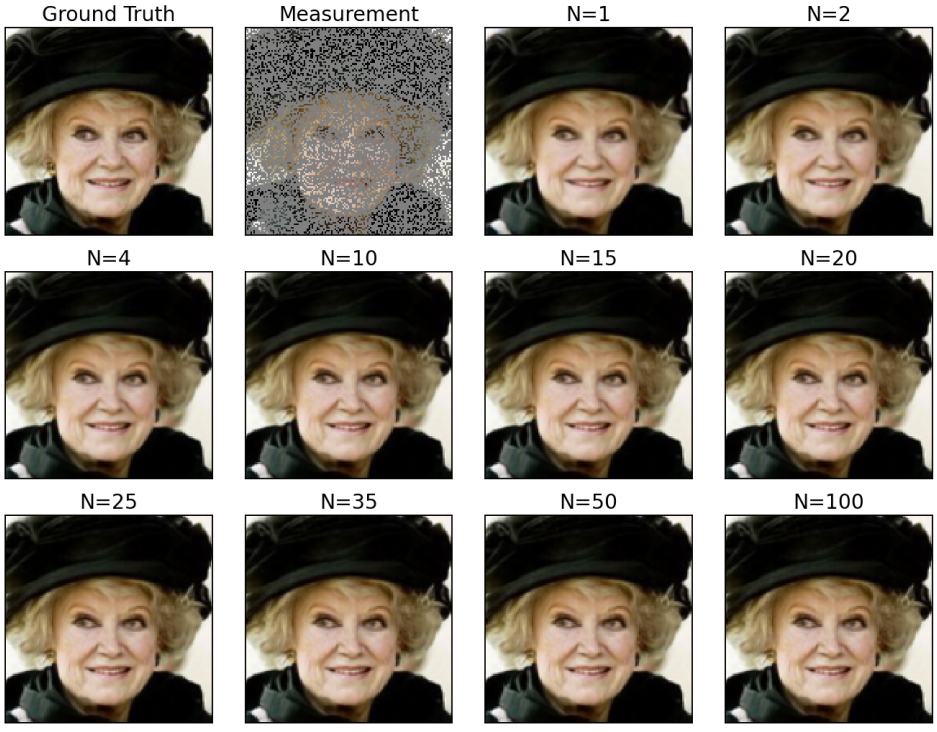

Figure 11: CelebA random inpainting: reconstructions as the number of sampling steps grows ($N = 1 \rightarrow 100$), showing added detail at higher N.

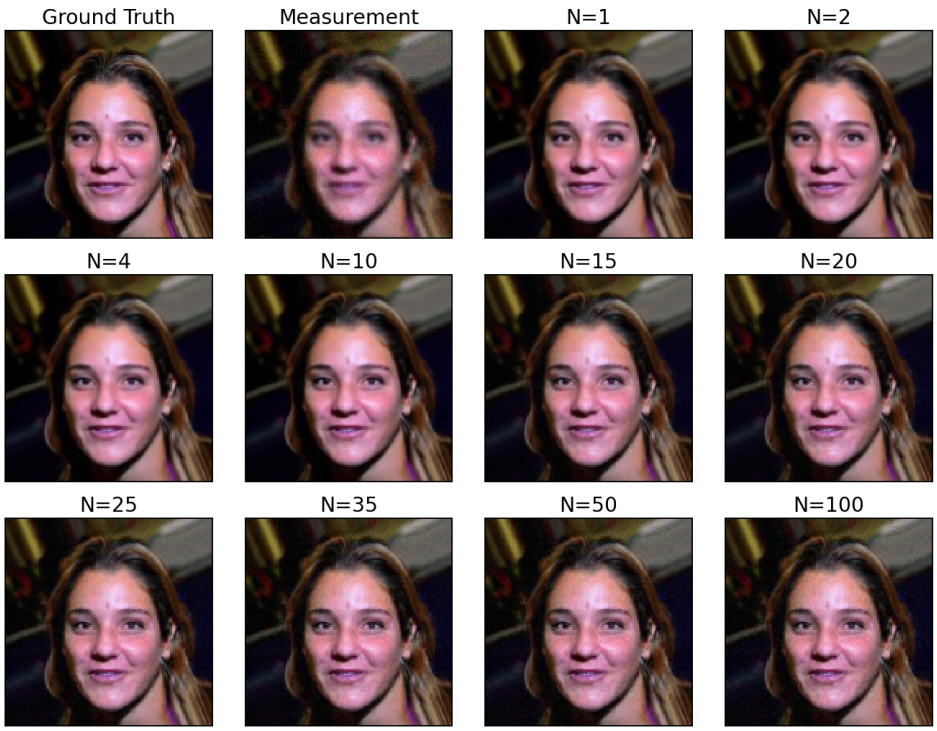

Figure 12: CelebA deblurring reconstructions across sampling steps ($N = 1 \rightarrow 100$).

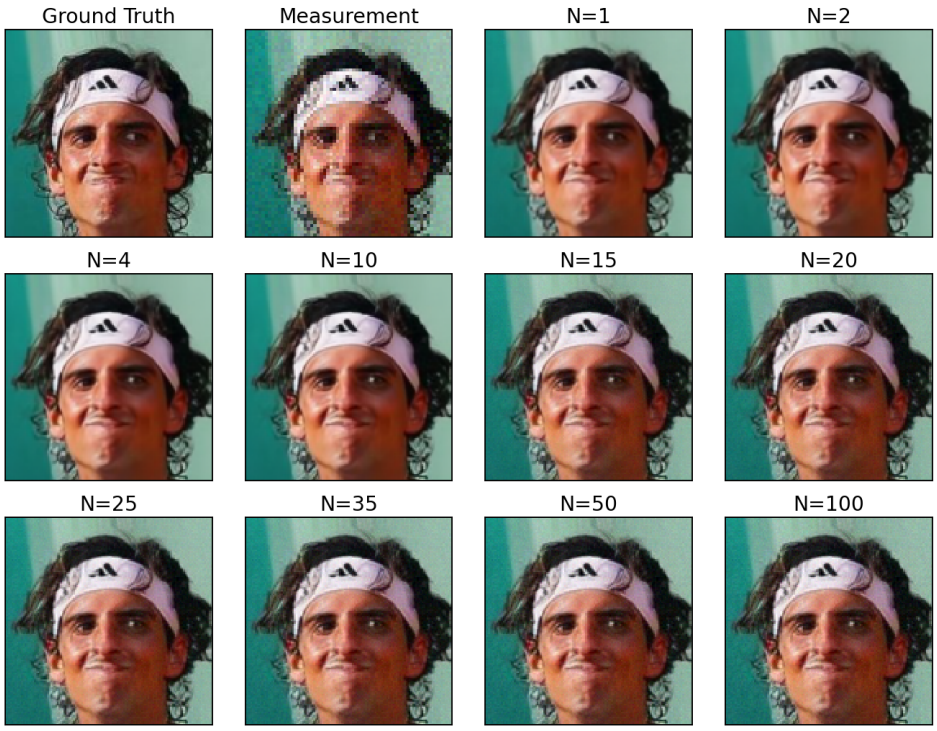

Figure 13: CelebA x2 super-resolution reconstructions across sampling steps ($N = 1 \rightarrow 100$).

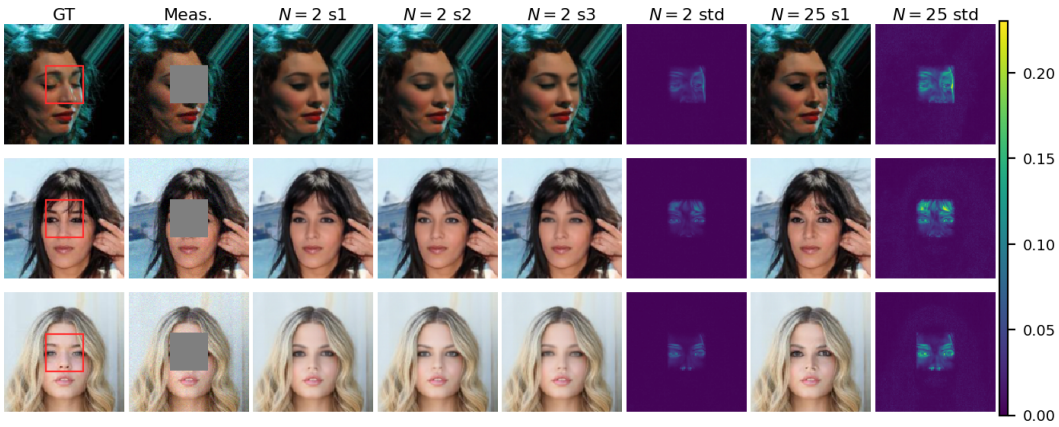

Figure 14: Posterior sample diversity on box inpainting (CelebA, 128×128, $K{=}5$). For each test image the measurement $\boldsymbol{y}$ is held fixed and only the source draw $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is varied. Columns show the ground truth, the measurement, three of the $S{=}8$ reconstructions at $N{=}2$, the pixel-wise standard deviation over all eight samples, and the corresponding sample and standard deviation at $N{=}25$. Standard-deviation maps share a single color scale across the figure. Variation is confined to the masked region and vanishes where measurements are available, consistent with Proposition 1, where the learned field transports the source to the measurement-conditioned posterior rather than to a single point estimate. Longer integration ($N{=}25$) widens the spread, which accounts for the improvement in LPIPS and the loss in PSNR reported in Table 1 and Table 11.

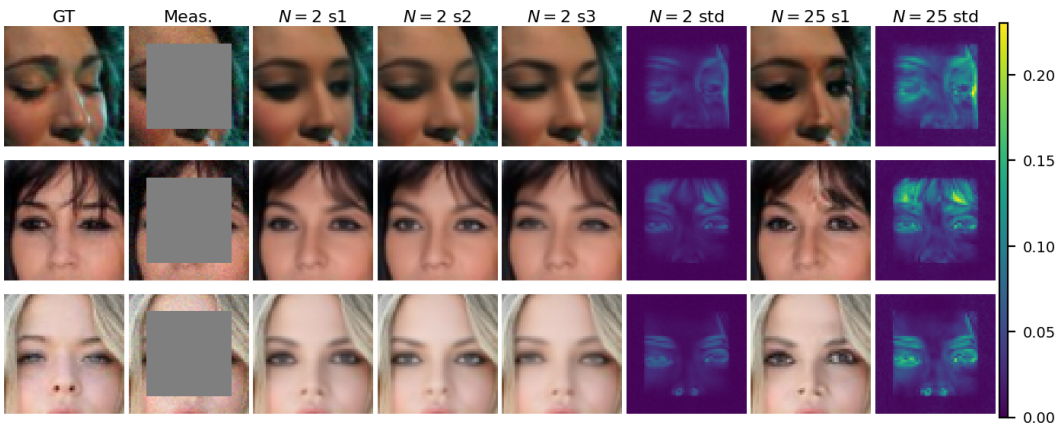

Figure 15: Zoomed-in version of Figure 14.

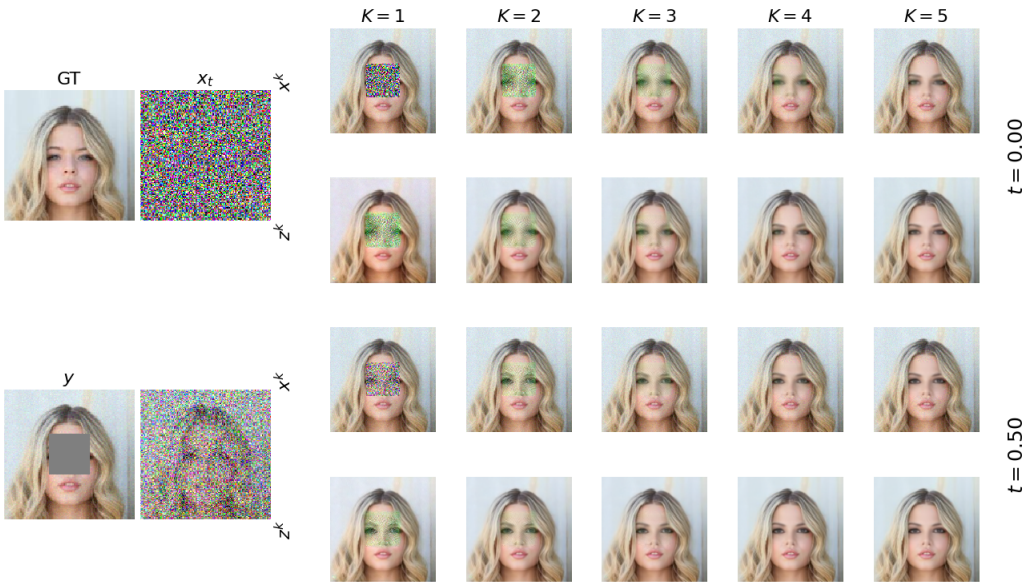

Figure 16: Internal trajectory of the solver for box inpainting ($N = 2$ Euler steps, $K = 5$ HQS iterations each): from the ground truth and measurement, each Euler step (rows, labeled by t) refines the estimate through its inner iterations $K = 1{:}5$, showing the data-consistency iterate $\boldsymbol{x}^k$ (top) and the damped learned update $\boldsymbol{z}^k$ (bottom).



Figure 17: Compressed-sensing Fourier sampling on AFHQ-Cat (256×256), $\sigma = 0.002$. Columns: ground truth, zero-filled adjoint $\boldsymbol{A}^*\boldsymbol{y}$, Flower1-OT, ours ($N$=2) with Cartesian mask (21.88%).