FLOW-RI: RESIDUAL INFERENCE WITH FLOW MATCHING FOR WIRELESS SPATIAL FIELD RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Inverse problems recover an unknown signal or physical field from incomplete and noisy observations. In wireless spatial field reconstruction, measurements are often available at only a small fraction of locations, making the problem particularly difficult at low sampling rates. A pretrained generative model can provide a strong prior for the complete field, while the local spatial correlation in wireless fields provides an additional structural prior that can be exploited through spatial regularization. A straightforward approach is to apply this regularization to the complete reconstructed field. However, doing so may alter structures that are already captured by the learned prior. To address this, we introduce Flow-RI, a residual inference framework built on a pretrained Flow Matching prior. Instead of regularizing the complete field, Flow-RI applies spatial regularization only to the prediction residual. At each flow step, the mismatch between the observed measurements and the predicted values at the sampled locations provides sparse observations of this residual. A spatial regularizer estimates the residual over the complete grid from these sparse observations, and the resulting residual estimate is used to correct the Flow Matching prediction before forming the next intermediate state. We further analyze the effect of applying spatial regularization to the prediction residual rather than to the complete field. At a fixed flow step, under the assumption that the reference prediction equals the conditional mean and that the measurement noise is conditionally zero-mean, residual regularization preserves conditional unbiasedness, whereas regularizing the complete field generally introduces an additional smoothing bias. Experiments on ray-traced wireless fields show that Flow-RI achieves the lowest mean absolute error (MAE) across all evaluated wireless reconstruction settings. The method also generalizes to different transmitter deployments and previously unseen urban scenes without retraining.

1 INTRODUCTION

Inverse problems seek to recover an unknown signal or physical field from incomplete or noisy observations. Examples arise in radio and optical interferometric imaging (McEwen and Wiaux, 2011; Roddier, 1988), seismic tomography (Rawlinson et al., 2014), and full-waveform inversion (Virieux and Operto, 2009). Because the measurements are often insufficient to determine the target uniquely, reconstruction requires prior information. Classical approaches impose explicit structure through sparsity, smoothness, or total variation (Rudin et al., 1992; Candès et al., 2006; Donoho, 2006; Ravishankar et al., 2020). More recently, pretrained generative models have provided data-driven priors that can be combined with the measurement model at inference time. This idea has been extensively studied with diffusion models (Song et al., 2021; Kawar et al., 2022; Chung et al., 2023) and has recently been extended to Flow Matching (Lipman et al., 2023; Zhang et al., 2024; Martin et al., 2025; Kim et al., 2025; Pourya et al., 2026).

A representative inverse problem in wireless systems is spatial field reconstruction, where quantities such as received signal strength, received power, or channel gain must be recovered over a region from measurements collected at only a small fraction of locations (Alaya-Feki et al., 2008; Romero and Kim, 2022; Zeng and Xu, 2021). At low sampling rates, large parts of the region remain unobserved, making reconstruction particularly dependent on prior information. Wireless fields also exhibit local spatial correlation, with nearby locations tending to be more strongly related. This structure provides a natural spatial prior and has motivated interpolation and neighborhood-based reconstruction methods

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.

(Bazerque et al., 2011; Romero and Kim, 2022; Zhang and Wang, 2022). Learned reconstruction networks (Levie et al., 2021; Teganya and Romero, 2022; Shrestha et al., 2022) and, more recently, generative priors (Fu et al., 2026; Sun et al., 2026) provide data-driven alternatives.

To exploit both learned field structure and local spatial correlation, we combine a pretrained generative prior with an explicit spatial prior. A straightforward way to combine them is to apply spatial regularization directly to the complete reconstructed field. However, doing so may also modify structures that are already captured by the learned prior. We instead apply the spatial regularizer only to the prediction residual, i.e., the difference between the unknown field and the current prediction. The mismatch between the observed measurements and the predicted values at the sampled locations provides sparse observations of this residual. In this way, the learned model provides the baseline field structure, while the spatial prior acts only on the remaining prediction error.

Based on this residual formulation, we introduce Flow-RI, a residual inference framework built on a pretrained Flow Matching prior. At each flow step, Flow-RI forms the measurement mismatch from the current Flow Matching prediction and estimates the corresponding residual over the complete grid using a local spatial regularizer. The estimated residual corrects the Flow Matching prediction, which is then used to construct the next flow state.

To analyze the difference between residual and complete-field regularization, we consider two estimators at a fixed flow step: one regularizes the prediction residual, whereas the other regularizes the complete field. We derive the corresponding error decompositions and show that, under the assumption that the reference prediction equals the conditional mean and that the measurement noise is conditionally zero-mean, residual regularization preserves conditional unbiasedness, whereas complete-field regularization generally introduces an additional smoothing bias.

We evaluate Flow-RI on ray-traced wireless fields across sampling rates, measurement noise, nonuniform sampling, and deployment and scene shifts. Flow-RI achieves the lowest mean absolute error (MAE) across all evaluated wireless reconstruction settings. The same frozen prior is used throughout these evaluations. Additional experiments in the appendix examine the residual-inference principle in natural image restoration.

Our contributions are:

- We introduce Flow-RI, which combines a pretrained Flow Matching prior with the local spatial prior of wireless fields by regularizing the prediction residual rather than the complete reconstructed field. The measurement mismatch provides sparse observations of this residual, which are used to estimate the correction over unobserved locations through spatial regularization.
- We compare residual regularization with complete-field regularization at a fixed flow step. Under the assumption that the reference prediction equals the conditional mean and that the measurement noise is conditionally zero-mean, the residual formulation preserves conditional unbiasedness, whereas complete-field regularization generally introduces an additional smoothing bias.
- We evaluate Flow-RI on sparse wireless spatial field reconstruction across sampling rates, measurement noise, sampling geometries, and deployment and scene shifts. Flow-RI achieves the lowest MAE across all evaluated wireless reconstruction settings. We further examine whether the residual-inference principle extends to natural image restoration tasks.

2 Related Work

Diffusion-based inverse solvers. Learned models have long been used as priors for ill-posed inverse problems. Early approaches either restricted the solution to signals represented by a learned model (Bora et al., 2017) or incorporated pretrained denoisers into iterative reconstruction algorithms (Venkatakrishnan et al., 2013; Romano et al., 2017; Ongie et al., 2020). More recently, diffusion models have become a widely used prior for inverse problems. Measurement information is incorporated during sampling in different ways, including spectral decomposition (Kawar et al., 2022), null-space updates (Wang et al., 2023), pseudoinverse-guided conditional scores (Song et al., 2023), posterior guidance (Chung et al., 2023), and plug-and-play reconstruction (Zhu et al., 2023; Shrestha and Fu, 2026).

Flow Matching and flow-based inverse solvers. Flow Matching learns a continuous transport from a source distribution to the data distribution through a time-dependent velocity field (Lipman et al., 2023). Related formulations include Rectified Flow, Multisample Flow Matching, and stochastic interpolants (Liu et al., 2023; Pooladian et al., 2023; Albergo et al., 2025). Several recent methods have adapted pretrained flow models to inverse problems. Flow-Priors constructs local inverse objectives along corrupted flow trajectories (Zhang et al., 2024); PnP-Flow alternates the flow prior with plug-and-play data-fidelity updates (Martin et al., 2025); D-Flow optimizes the source sample by differentiating through the flow (Ben-Hamu et al., 2024); and FlowDPS incorporates posterior guidance during flow-based sampling (Kim et al., 2025). FlowChef develops inference-time steering for rectified flows (Patel et al., 2025), while FLOWER derives a measurement-aware proximal update from an isotropic Gaussian approximation of the conditional endpoint distribution (Pourya et al., 2026). Flow-RI differs from these methods in how measurement information is incorporated. It exploits the local spatial correlation of wireless fields by treating the mismatch between the measurements and the Flow Matching prediction as sparse observations of the prediction residual. Spatial regularization then estimates this residual at unobserved locations and uses it to correct the Flow Matching prediction.

Wireless spatial field reconstruction. Wireless spatial fields have traditionally been estimated from geographically distributed measurements using interpolation (Alaya-Feki et al., 2008), spline-based reconstruction (Bazerque et al., 2011), kernel methods (Romero and Kim, 2022), and neighborhood-based estimators (Zhang and Wang, 2022). Learned approaches include RadioUNet (Levie et al., 2021), deep completion autoencoders (Teganya and Romero, 2022), and neural wireless spatial field completion models (Shrestha et al., 2022), while latent-domain plug-and-play denoising has also been developed for wireless spatial field estimation (Xu et al., 2026). Continuous neural representations have also been explored for wireless propagation fields, including NeRF^2 (Zhao et al., 2023), NeWRF (Lu et al., 2024), and WRF-GS (Wen et al., 2025). More recently, generative models have been explored for wireless field reconstruction. CKMDiff develops a conditional diffusion model for wireless field reconstruction (Fu et al., 2026), while Flow Matching has been combined with unmanned aerial vehicle (UAV) active sampling for wireless spatial field construction (Sun et al., 2026). NeWRF and WRF-GS learn representations of a particular wireless environment from measurements, while CKMDiff uses task-specific conditional diffusion for wireless field reconstruction. In contrast, Flow-RI uses a frozen Flow Matching prior trained only on complete wireless fields and incorporates sparse measurements only at inference time.

3 Problem Formulation and Flow Matching Prior

In this section, we formulate the sparse wireless spatial field reconstruction problem and introduce the pretrained Flow Matching model used as the learned prior. The sparse measurements provide observations of the unknown field at only a subset of locations, while the Flow Matching model provides a prediction over the complete field.

3.1 Sparse Wireless Spatial Field Reconstruction

Consider a two-dimensional region $\mathcal{S} \subset \mathbb{R}^2$ over which the wireless field is to be reconstructed. We discretize $\mathcal{S}$ into an $H \times W$ grid and let $P = H \times W$ denote the total number of grid locations. The unknown field is represented by $\mathbf{X}^\star \in \mathbb{R}^{H \times W}$, where $\mathbf{X}^\star_{[i,j]}$ denotes the field value at grid location (i, j). Depending on the application, $\mathbf{X}^\star$ may represent received signal strength, received power, or another spatially varying wireless quantity (Romero and Kim, 2022).

Measurements are collected at M grid locations. Let

$$\Omega = \{(i_m, j_m)\}_{m=1}^{M} \subseteq \{1, \dots, H\} \times \{1, \dots, W\}$$

denote the set of sampled locations. The measurement collected at (i_m, j_m) is

$$y_m = \mathbf{X}^\star_{[i_m, j_m]} + n_m, \qquad m = 1, \dots, M, \tag{1}$$

where $n_m \sim \mathcal{N}(0, \sigma^2)$ denotes independent and identically distributed (i.i.d.) Gaussian measurement noise at the sampled location (i_m, j_m).

Let $\mathrm{vec} : \mathbb{R}^{H\times W} \to \mathbb{R}^P$ denote a fixed vectorization operator, and let $\mathbf{y} = [y_1, \ldots, y_M]^\top$ and $\mathbf{n} = [n_1, \ldots, n_M]^\top$ collect the measurements and noise, respectively. The measurement model can then be written as

$$\mathbf{y} = \mathcal{A}_\Omega \,\mathrm{vec}(\mathbf{X}^\star) + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_M), \tag{2}$$

where $\mathcal{A}_\Omega \in \{0,1\}^{M\times P}$ is the sampling matrix associated with Ω, with exactly one nonzero entry in each row indicating the corresponding sampled grid location, and $\mathbf{I}_M$ is the $M \times M$ identity matrix.

Our goal is to recover the complete field $\mathbf{X}^\star$ from $\mathbf{y}$ and Ω. Since $M \ll P$, the observations alone do not determine the values at the unobserved locations. We therefore use a pretrained Flow Matching model (Lipman et al., 2023) as a learned prior over complete wireless fields.

3.2 Flow Matching Prior

Flow Matching formulation. Flow Matching learns a time-dependent velocity field that transports samples from a source distribution p_0 to the data distribution p_1. In our setting, p_1 is the distribution of complete wireless fields, while p_0 is a standard Gaussian distribution. Let $\mathbf{X}_0 \sim p_0$ and $\mathbf{X}_1 \sim p_1$, where $\mathbf{X}_0, \mathbf{X}_1 \in \mathbb{R}^{H\times W}$.

We use the linear interpolation

$$\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{X}_1, \qquad t \in [0,1]. \tag{3}$$

Differentiating this path with respect to t gives the training target $\mathbf{X}_1 - \mathbf{X}_0$. The velocity field $\mathbf{v}_\theta(\mathbf{X}_t, t)$ is trained to predict this target from the intermediate state $\mathbf{X}_t$ and time t by minimizing

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, \mathbf{X}_0\sim p_0,\, \mathbf{X}_1\sim p_1} \left[\|\mathbf{v}_\theta(\mathbf{X}_t, t) - (\mathbf{X}_1 - \mathbf{X}_0)\|_F^2 \right]. \tag{4}$$

After training, generation follows the ordinary differential equation (ODE)

$$\frac{\mathrm{d}\mathbf{X}_t}{\mathrm{d}t} = \mathbf{v}_\theta(\mathbf{X}_t, t), \qquad t \in [0,1]. \tag{5}$$

Starting from $\mathbf{X}_0 \sim p_0$, integrating this ODE transports the initial sample toward the data distribution p_1.

Endpoint prediction. The Flow Matching model is trained to predict the velocity $\mathbf{X}_1 - \mathbf{X}_0$, whereas our reconstruction method requires an estimate of the endpoint $\mathbf{X}_1$. For the linear interpolation in Equation (3), rearranging the interpolation gives

$$\mathbf{X}_1 = \mathbf{X}_t + (1-t)(\mathbf{X}_1 - \mathbf{X}_0). \tag{6}$$

Under the squared-error objective in Equation (4), the minimum mean squared error (MSE) predictor of the velocity target is

$$\mathbf{v}^\star(\mathbf{X}_t, t) = \mathbb{E}\left[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_t, t\right]. \tag{7}$$

Applying the conditional expectation $\mathbb{E}[\cdot \mid \mathbf{X}_t, t]$ to Equation (6) and using Equation (7) gives

$$\mathbb{E}\left[\mathbf{X}_1 \mid \mathbf{X}_t, t\right] = \mathbf{X}_t + (1-t)\mathbf{v}^\star(\mathbf{X}_t, t). \tag{8}$$

The learned velocity field $\mathbf{v}_\theta$ approximates $\mathbf{v}^\star$. Replacing $\mathbf{v}^\star$ by $\mathbf{v}_\theta$ therefore gives the endpoint prediction

$$\widehat{\mathbf{X}}_{1,t} = \mathbf{X}_t + (1-t)\mathbf{v}_\theta(\mathbf{X}_t, t). \tag{9}$$

During reconstruction, $\widehat{\mathbf{X}}_{1,t}$ serves as the reference prediction. The next section uses its mismatch with the sparse measurements to infer a spatially regularized residual correction.

4 Flow Residual Inference

In this section, we develop Flow-RI using the pretrained Flow Matching model introduced above. At each flow time, we take the current target prediction as a reference and use its mismatch with the measurements to estimate the remaining prediction error. The corrected target is then used to update the flow state.

4.1 Residual Estimation from Sparse Measurements

At a given flow time t, define the prediction residual as

$$\mathbf{R}_t^\star = \mathbf{X}^\star - \widehat{\mathbf{X}}_{1,t}, \tag{10}$$

so that $\mathbf{X}^\star = \widehat{\mathbf{X}}_{1,t} + \mathbf{R}_t^\star$. Subtracting the predicted measurements from the observations gives

$$\mathbf{e}_t = \mathbf{y} - \mathcal{A}_\Omega \operatorname{vec}(\widehat{\mathbf{X}}_{1,t}) = \mathcal{A}_\Omega \operatorname{vec}(\mathbf{R}_t^\star) + \mathbf{n}. \tag{11}$$

Thus, $\mathbf{e}_t$ is the prediction residual observed on Ω, corrupted by measurement noise.

We model the remaining prediction error as locally correlated and apply the spatial prior only to this residual, leaving the Flow Matching prediction itself unregularized. Since the residual is observed only at the sampled locations, we use local spatial regularization to estimate it elsewhere. Let $\mathcal{E} \subseteq \big\{\{\ell, s\} : 1 \le \ell < s \le P\big\}$ denote the set of unordered pairs of spatial locations connected under a prescribed local neighborhood relation. For each $\{\ell, s\} \in \mathcal{E}$, let $w_{\ell s} > 0$ denote the corresponding pairwise weight. With $R_\ell = [\operatorname{vec}(\mathbf{R})]_\ell$, we define

$$\mathcal{R}(\mathbf{R}) = \frac{1}{2} \sum_{\{\ell,s\}\in\mathcal{E}} w_{\ell s}(R_\ell - R_s)^2. \tag{12}$$

This regularizer encourages the residual to vary smoothly between nearby locations.

For convenience, Equation (12) can be written in quadratic form as

$$\mathcal{R}(\mathbf{R}) = \frac{1}{2} \operatorname{vec}(\mathbf{R})^\top \mathbf{L} \operatorname{vec}(\mathbf{R}), \tag{13}$$

where $\mathbf{L} \in \mathbb{R}^{P\times P}$ is the symmetric positive semidefinite spatial regularization matrix determined by $\mathcal{E}$ and the weights $\{w_{\ell s}\}$.

The complete residual is then estimated by

$$\widehat{\mathbf{R}}_t = \arg\min_{\mathbf{R}\in\mathbb{R}^{H\times W}} \frac{1}{2} \|\mathcal{A}_\Omega \operatorname{vec}(\mathbf{R}) - \mathbf{e}_t\|_2^2 + \frac{\lambda}{2} \operatorname{vec}(\mathbf{R})^\top \mathbf{L} \operatorname{vec}(\mathbf{R}), \tag{14}$$

where $\lambda > 0$ controls the regularization strength. The first term fits the residual to the observed prediction mismatch, while the second penalizes large differences between locally related residual values.

The first-order optimality condition of Equation (14) is

$$\big(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L}\big) \operatorname{vec}(\widehat{\mathbf{R}}_t) = \mathcal{A}_\Omega^\top \mathbf{e}_t. \tag{15}$$

If the chosen local pairwise relations connect all spatial locations, then $\ker(\mathbf{L}) = \operatorname{span}\{\mathbf{1}\}$. Since at least one location is observed, $\mathcal{A}_\Omega \mathbf{1} \neq \mathbf{0}$. Therefore, $\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L}$ is positive definite for $\lambda > 0$, and the residual estimate is unique. Equivalently,

$$\operatorname{vec}(\widehat{\mathbf{R}}_t) = \big(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L}\big)^{-1} \mathcal{A}_\Omega^\top \mathbf{e}_t. \tag{16}$$

4.2 Residual-Corrected Flow Update

We next incorporate the residual estimate into the Flow Matching update. Let $0 = t_0 < t_1 < \cdots < t_N = 1$ denote the discrete time grid. Reconstruction starts from $\mathbf{Z}_0 \sim p_0$ with $\mathbf{X}_{t_0} = \mathbf{Z}_0$.

At step k, Equation (9) gives the target prediction $\widehat{\mathbf{X}}_{1,t_k}$ from the current state $\mathbf{X}_{t_k}$. We then form $\mathbf{e}_{t_k}$ using Equation (11) and estimate $\widehat{\mathbf{R}}_{t_k}$ from Equation (15). The corrected target is

$$\widetilde{\mathbf{X}}_{1,t_k} = \widehat{\mathbf{X}}_{1,t_k} + \widehat{\mathbf{R}}_{t_k}. \tag{17}$$

In this update, spatial regularization is applied to the correction rather than to the predicted field itself.

To construct the next state, we draw a fresh source sample $\mathbf{Z}_{k+1} \sim p_0$ and combine it with the current corrected target. Using the same source–target interpolation form as in Equation (3), we set

$$\mathbf{X}_{t_{k+1}} = (1 - t_{k+1})\mathbf{Z}_{k+1} + t_{k+1}\widetilde{\mathbf{X}}_{1,t_k}. \tag{18}$$

The updated state is passed through the pretrained model again to obtain a new target prediction and residual estimate. This procedure is repeated until $t_N = 1$, for which $\mathbf{X}_{t_N} = \widetilde{\mathbf{X}}_{1,t_{N-1}}$ is returned as the reconstructed field.

4.3 Algorithm and Computation

Algorithm 1 summarizes the reconstruction procedure.

Algorithm 1 Flow-RI reconstruction from sparse noisy measurements

Require: observations $\mathbf{y}$, sampling matrix $\mathcal{A}_\Omega$, spatial regularization matrix $\mathbf{L}$, regularization parameter λ, pretrained velocity field $\mathbf{v}_\theta$, and time grid $\{t_k\}_{k=0}^{N}$

$\mathbf{Z}_0 \sim p_0, \qquad \mathbf{X}_{t_0} \leftarrow \mathbf{Z}_0$

for $k = 0, \dots, N-1$ **do**

$\widehat{\mathbf{X}}_{1,t_k} \leftarrow \mathbf{X}_{t_k} + (1-t_k)\mathbf{v}_\theta(\mathbf{X}_{t_k}, t_k)$ ▷ Predict target

$\mathbf{e}_{t_k} \leftarrow \mathbf{y} - \mathcal{A}_\Omega \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k})$ ▷ Form residual observations

$\operatorname{vec}(\widehat{\mathbf{R}}_{t_k}) \leftarrow \left(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L}\right)^{-1} \mathcal{A}_\Omega^\top \mathbf{e}_{t_k}$ ▷ Estimate residual

$\widetilde{\mathbf{X}}_{1,t_k} \leftarrow \widehat{\mathbf{X}}_{1,t_k} + \widehat{\mathbf{R}}_{t_k}$ ▷ Correct target

$\mathbf{X}_{t_{k+1}} \leftarrow (1-t_{k+1})\mathbf{Z}_{k+1} + t_{k+1}\widetilde{\mathbf{X}}_{1,t_k}, \qquad \mathbf{Z}_{k+1} \sim p_0$ ▷ Update flow state

return $\mathbf{X}_{t_N}$

The coefficient matrix $\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L}$ is unchanged across flow steps because $\mathcal{A}_\Omega$, $\mathbf{L}$, and λ remain fixed. The matrix and the associated solver components can therefore be prepared once and reused across iterations; only the right-hand side $\mathcal{A}_\Omega^\top \mathbf{e}_{t_k}$ changes. Each iteration of Flow-RI requires one Flow Matching evaluation and one residual solve.

5 Analysis of Residual Regularization

In this section, we compare two ways of combining the current Flow Matching prediction with spatial regularization. We first examine how each formulation corrects the prediction, then compare the conditional biases of the resulting estimators at a fixed flow step.

5.1 Residual versus Complete-Field Regularization

We treat the current Flow Matching prediction $\widehat{\mathbf{X}}_{1,t_k}$ as fixed throughout this analysis.

Writing $\mathbf{Z} = \widehat{\mathbf{X}}_{1,t_k} + \mathbf{R}$, the residual formulation in Equation (14) can be expressed directly in terms of the reconstructed field:

$$\begin{aligned}\widetilde{\mathbf{X}}_{1,t_k} = \arg\min_{\mathbf{Z}\in\mathbb{R}^{H\times W}} \; & \frac{1}{2}\left\|\mathcal{A}_\Omega \operatorname{vec}(\mathbf{Z}) - \mathbf{y}\right\|_2^2 \\ & + \frac{\lambda}{2} \operatorname{vec}\Big(\mathbf{Z} - \widehat{\mathbf{X}}_{1,t_k}\Big)^\top \mathbf{L} \operatorname{vec}\Big(\mathbf{Z} - \widehat{\mathbf{X}}_{1,t_k}\Big).\end{aligned} \tag{19}$$

Hence, the spatial penalty is imposed on the correction $\mathbf{Z} - \widehat{\mathbf{X}}_{1,t_k}$, while the Flow Matching prediction itself is not spatially regularized.

We compare this with a complete-field formulation that applies the same spatial penalty directly to $\mathbf{Z}$. To keep the current Flow Matching prediction as a reference, we include a quadratic reference term:

$$\begin{aligned}\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k} = \arg\min_{\mathbf{Z}\in\mathbb{R}^{H\times W}} \; & \frac{1}{2}\left\|\mathcal{A}_\Omega \operatorname{vec}(\mathbf{Z}) - \mathbf{y}\right\|_2^2 + \frac{\lambda}{2} \operatorname{vec}(\mathbf{Z})^\top \mathbf{L} \operatorname{vec}(\mathbf{Z}) \\ & + \frac{\gamma}{2}\left\|\mathbf{Z} - \widehat{\mathbf{X}}_{1,t_k}\right\|_F^2,\end{aligned} \tag{20}$$

where $\gamma > 0$ controls the strength of the reference term.

The distinction between the two formulations becomes clearer when their corrections to $\widehat{\mathbf{X}}_{1,t_k}$ are written explicitly. For residual regularization, the correction is driven by $\mathcal{A}_\Omega^\top \mathbf{e}_{t_k}$ through Equation (15). For complete-field regularization, the corresponding correction satisfies

$$\left(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\right) \operatorname{vec}\Big(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k} - \widehat{\mathbf{X}}_{1,t_k}\Big) = \mathcal{A}_\Omega^\top \mathbf{e}_{t_k} - \lambda \mathbf{L} \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k}), \tag{21}$$

where $\mathbf{I}_P$ is the $P \times P$ identity matrix. Unlike the residual formulation, this correction contains the additional term $-\lambda \mathbf{L} \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k})$, which comes from regularizing the complete field.

If the current prediction already matches the observed measurements, i.e., $\mathbf{e}_{t_k} = \mathbf{0}$, residual regularization produces no correction: $\widehat{\mathbf{R}}_{t_k} = \mathbf{0}$ and $\widetilde{\mathbf{X}}_{1,t_k} = \widehat{\mathbf{X}}_{1,t_k}$. In contrast, complete-field regularization may still modify the prediction because the spatial penalty acts directly on the field. The prediction is left unchanged only when $\mathbf{L} \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k}) = \mathbf{0}$.

5.2 Fixed-Step Conditional Bias Analysis

To compare the estimation errors for the two regularization formulations, define $\mathcal{T}_\lambda = \left(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L}\right)^{-1} \mathcal{A}_\Omega^\top$, and $\mathcal{B}_\lambda = \mathbf{I}_P - \mathcal{T}_\lambda \mathcal{A}_\Omega$. Substituting the residual estimate in Equation (16) into the corrected target in Equation (17), and using Equation (11), gives

$$\operatorname{vec}(\widetilde{\mathbf{X}}_{1,t_k}) = \mathcal{T}_\lambda \mathbf{y} + \mathcal{B}_\lambda \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k}). \tag{22}$$

Substituting the observation model in Equation (2) gives

$$\operatorname{vec}\left(\widetilde{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\right) = \mathcal{B}_\lambda \operatorname{vec}\left(\widehat{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\right) + \mathcal{T}_\lambda \mathbf{n}. \tag{23}$$

Thus, the residual-regularized error depends on the prediction error and the propagated measurement noise.

For the complete-field-regularized estimator, the corresponding error is

$$\begin{aligned} \operatorname{vec}\left(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k} - \mathbf{X}^\star\right) = {} & \left(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\right)^{-1} \\ & \times \left[\gamma \operatorname{vec}\left(\widehat{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\right) - \lambda \mathbf{L} \operatorname{vec}(\mathbf{X}^\star) + \mathcal{A}_\Omega^\top \mathbf{n}\right]. \end{aligned} \tag{24}$$

The derivations of the two error decompositions in Equations (23) and (24) are given in Appendix A. Compared with Equation (23), the complete-field estimator contains the additional term $-\lambda \mathbf{L} \operatorname{vec}(\mathbf{X}^\star)$, which arises from applying the spatial penalty directly to the field.

We next compare the conditional biases of the two estimators. Treat $\mathbf{X}^\star$ as a random target field with finite second moments, and define

$$\boldsymbol{\mu}_{t_k} = \mathbb{E}[\mathbf{X}^\star \mid \mathbf{X}_{t_k}, t_k].$$

Under the training interpolation, Equation (8) motivates the Flow Matching prediction as a conditional-mean estimate. During reconstruction, however, $\mathbf{X}_{t_k}$ is repeatedly updated using the measurements, so the same conditional-mean relation need not hold along the inference trajectory. Moreover, because the state depends on the measurements, the measurement noise need not have zero conditional mean given the current state. We therefore state the following fixed-step result under these two conditions explicitly.

Proposition 5.1 (Conditional Bias Comparison)**.** *Suppose that, at flow step k,*

$$\widehat{\mathbf{X}}_{1,t_k} = \boldsymbol{\mu}_{t_k}, \qquad \mathbb{E}[\mathbf{n} \mid \mathbf{X}_{t_k}, t_k] = \mathbf{0}.$$

Then the residual-regularized estimator satisfies

$$\mathbb{E}\left[\operatorname{vec}\left(\widetilde{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\right) \mid \mathbf{X}_{t_k}, t_k\right] = \mathbf{0}. \tag{25}$$

For any fixed $\gamma > 0$, the complete-field-regularized estimator satisfies

$$\begin{aligned} & \mathbb{E}\left[\operatorname{vec}\left(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k} - \mathbf{X}^\star\right) \mid \mathbf{X}_{t_k}, t_k\right] \\ & \quad = -\left(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\right)^{-1} \lambda \mathbf{L} \operatorname{vec}(\boldsymbol{\mu}_{t_k}). \end{aligned} \tag{26}$$

Proof. See Appendix A.

The first assumption does not require the prediction to equal $\mathbf{X}^\star$ for every realization; it only requires the prediction to equal the conditional mean at the current flow step. Under the two conditions in Proposition 5.1, residual regularization is conditionally unbiased. In contrast, for fixed $\lambda > 0$ and any finite $\gamma > 0$, the complete-field estimator is conditionally biased whenever $\mathbf{L} \operatorname{vec}(\boldsymbol{\mu}_{t_k}) \neq \mathbf{0}$. This comparison concerns conditional bias only and does not imply lower reconstruction error in every realization, since the error variances may differ. Their end-to-end reconstruction performance is compared in Section 6.2.

6 EXPERIMENTS

We evaluate Flow-RI on ray-traced wireless fields, focusing on the regularization target and reconstruction accuracy under sparse sampling. Appendix D reports noise, sampling-geometry, transfer, and runtime results. Appendix E evaluates the same residual update on natural-image restoration tasks.

6.1 EXPERIMENTAL SETUP

Data and learned prior. We generate complete wireless spatial fields using the ray-tracing (RT) module Sionna RT (Aït Aoudia et al., 2025) in the open-source Etoile urban scene. The receiver plane spans $863.663 \times 686.121\ \mathrm{m}^2$ at a height of 60 m and is discretized on a 128×128 grid. Each field corresponds to an independently sampled five-transmitter deployment, with each grid cell recording the strongest received signal among the five links. The dataset contains 2,000 training fields, 100 validation fields, and 100 test fields. The Flow Matching model is trained only on complete training fields and remains frozen during reconstruction. Data-generation and training details are provided in Appendix B.

Baselines. We compare Flow-RI with inverse-distance-weighted K-nearest-neighbor (KNN) interpolation and five Flow Matching-based inverse solvers: DPS-ODE, PnP-Flow (Martin et al., 2025), FlowDPS (Kim et al., 2025), FlowChef (Patel et al., 2025), and FLOWER (Pourya et al., 2026). DPS-ODE is our deterministic ODE-based adaptation of diffusion posterior sampling (DPS) (Chung et al., 2023). All Flow Matching-based methods use the same frozen pretrained prior. Appendix C gives the implementation settings and validation ranges.

Evaluation. We report mean absolute error (MAE), normalized mean squared error (NMSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) (Wang et al., 2004) over the complete 128×128 field. Results are averaged over all 100 test fields. All methods use the same observation masks and noisy measurements.

6.2 EFFECT OF THE REGULARIZATION TARGET

We compare Flow-RI with Flow-CR, the complete-field counterpart defined in Equation (20). Flow-CR applies spatial regularization to the reconstructed field, whereas Flow-RI applies it to the prediction residual. For a fair comparison, both methods use the same $\mathbf{L}$, constructed from the four-neighbor structure with unit weights. Test fields, observation masks, measurement noise, and feedback-source sequences are paired across methods. Details of hyperparameter selection are given in Appendix C.3.

Residual regularization gives lower MAE at all four sampling rates (Table 1). The difference from Flow-CR decreases from 1.945 dB at 1% sampling to 1.070, 0.430, and 0.180 dB at 2%, 5%, and 10%, respectively. The gap is largest at low sampling rates and narrows as more measurements become available, even with γ selected independently for Flow-CR.

Figure 1 shows the corresponding reconstructions for a fixed test field. At 1% sampling, the Flow-CR reconstruction is visibly smoother and loses more fine-scale variation around the high-signal regions, whereas Flow-RI retains more of this structure. Their MAEs are 7.42 and 5.80 dB, respectively. At 10% sampling, the two reconstructions are more similar, with MAEs of 3.85 and 3.70 dB.

Table 1: Comparison of residual and complete-field regularization.

Method	1%	2%	5%	10%
Flow-CR	7.963	6.162	4.652	3.856
Flow-RI	**6.018**	**5.092**	**4.222**	**3.676**

We also compare Flow-RI with three component ablations that isolate the roles of the learned prior, spatial propagation of the residual, and intermediate residual feedback: spatial-only reconstruction without the learned prior, pointwise residual correction without spatial coupling, and terminal-only correction without intermediate residual feedback. All three ablations perform worse than Flow-RI across the evaluated sampling rates (Table 7 in Appendix C.4).

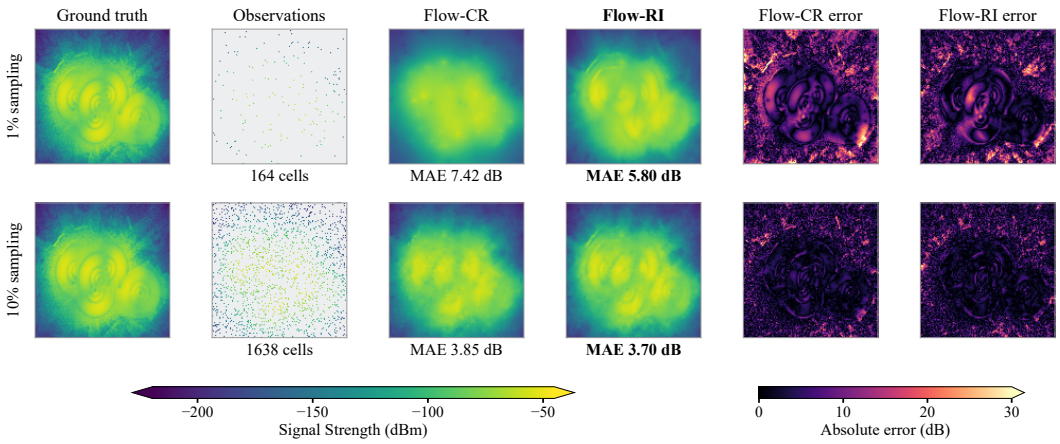


Figure 1: Qualitative comparison of residual and complete-field regularization at 1% (top) and 10% (bottom) sampling with 3 dB measurement noise. Columns show the ground-truth field, sparse observations, Flow-CR, Flow-RI, and their absolute errors.

6.3 Reconstruction Performance Across Sampling Rates

We next compare Flow-RI with the six baselines at 1%, 2%, 5%, and 10% uniform sampling with 3 dB measurement noise.

Flow-RI improves most strongly in the sparse regime targeted by the method. The MAE gain over the strongest baseline is 0.783 and 0.881 dB at 1% and 2% sampling, respectively, and becomes smaller as the sampling rate increases. Qualitative comparisons at 1% and 10% sampling are provided in Appendix D.1.

Table 2: Reconstruction metrics under uniform sampling with 3 dB measurement noise.

1% sampling					2% sampling				
Method	MAE ↓	NMSE ↓	PSNR ↑	SSIM ↑	Method	MAE ↓	NMSE ↓	PSNR ↑	SSIM ↑
KNN	8.414	0.0094	23.378	0.629	KNN	6.560	0.0056	25.605	0.687
DPS-ODE	6.840	0.0064	24.999	0.659	DPS-ODE	6.304	0.0056	26.045	0.696
PnP-Flow	10.929	0.0146	21.698	0.622	PnP-Flow	7.452	0.0068	24.860	0.666
FlowDPS	7.385	0.0067	24.888	0.662	FlowDPS	5.975	0.0045	26.590	0.697
FlowChef	13.120	0.0205	20.286	0.530	FlowChef	8.731	0.0090	23.592	0.582
FLOWER	16.196	0.0293	18.688	0.575	FLOWER	11.304	0.0147	21.571	0.615
Flow-RI	**6.057**	**0.0045**	**26.528**	**0.720**	Flow-RI	**5.094**	**0.0032**	**27.954**	**0.744**
5% sampling					10% sampling				
Method	MAE ↓	NMSE ↓	PSNR ↑	SSIM ↑	Method	MAE ↓	NMSE ↓	PSNR ↑	SSIM ↑
KNN	4.945	0.0031	28.069	0.734	KNN	4.110	0.0022	29.685	0.773
DPS-ODE	4.737	0.0030	28.492	0.748	DPS-ODE	3.958	0.0020	29.936	0.789
PnP-Flow	5.265	0.0099	26.173	0.701	PnP-Flow	4.227	0.0024	29.308	0.765
FlowDPS	4.745	0.0029	28.382	0.742	FlowDPS	4.042	0.0022	29.664	0.777
FlowChef	6.202	0.0047	26.302	0.642	FlowChef	5.233	0.0034	27.649	0.680
FLOWER	6.237	0.0047	26.336	0.688	FLOWER	4.586	0.0027	28.699	0.752
Flow-RI	**4.213**	**0.0023**	**29.502**	**0.777**	Flow-RI	**3.657**	**0.0017**	**30.687**	**0.806**

7 Conclusion

We introduced Flow-RI for reconstructing wireless fields from sparse measurements using a Flow Matching prior. It estimates a spatially regularized prediction residual from the measurement mismatch and feeds the corrected target back into the flow trajectory. At a fixed step, an exact conditional-mean reference and conditionally zero-mean measurement noise yield conditional unbiasedness for residual regularization, whereas complete-field regularization generally introduces smoothing bias. Across the evaluated wireless reconstruction settings, Flow-RI attains the lowest MAE and transfers across deployments and scenes without retraining.

Reproducibility Statement

The proofs of the theoretical results are provided in Appendix A. Appendix B describes the data generation, pretrained model, normalization, spatial prior, numerical solver, and inference settings. Appendix C specifies the evaluation protocol, including paired evaluation, hyperparameter selection, and evaluation metrics. Appendix D reports the additional robustness, generalization, and runtime experiments. The evaluation code and configuration files are provided in the anonymous supplementary material.

AI Use Statement

Generative AI tools were used to assist with the implementation of the proposed method and several baseline methods, and to support the development and checking of mathematical proofs. All AI-assisted code was reviewed and tested by the authors, and all mathematical arguments and proofs were independently verified. The authors take full responsibility for the final content, implementation, and results of this work.

References

3rd Generation Partnership Project (3GPP). Study on Channel Model for Frequencies from 0.5 to 100 GHz. Technical Report 38.901, version 19.4.0, 2026.

Fayçal Aït Aoudia, Jakob Hoydis, Merlin Nimier-David, Baptiste Nicolet, Sebastian Cammerer, and Alexander Keller. Sionna RT: Technical Report. *arXiv preprint arXiv:2504.21719*, 2025.

Afef Ben Hadj Alaya-Feki, Sana Ben Jemaa, Berna Sayraç, Paul Houzé, and Eric Moulines. Informed Spectrum Usage in Cognitive Radio Networks: Interference Cartography. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2008.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Juan Andrés Bazerque, Gonzalo Mateos, and Georgios B. Giannakis. Group-Lasso on Splines for Spectrum Cartography. *IEEE Transactions on Signal Processing*, 59(10):4648–4663, 2011.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through Flows for Controlled Generation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 3462–3483, 2024.

Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. Compressed Sensing Using Generative Models. In *Proceedings of the 34th International Conference on Machine Learning*, pages 537–546, 2017.

Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *International Conference on Learning Representations*, 2023.

David L. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

Shen Fu, Yong Zeng, Zijian Wu, Di Wu, Jie Xu, Shi Jin, Cheng-Xiang Wang, and Xiqi Gao. CKMDiff: A Generative Diffusion Model for CKM Construction via Inverse Problems with Learned Priors. *npj Wireless Technology*, 2:35, 2026.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising Diffusion Restoration Models. In *Advances in Neural Information Processing Systems*, volume 35, pages 23593–23606, 2022.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-Driven Posterior Sampling for Inverse Problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12328–12337, 2025.

Ron Levie, Çağkan Yapar, Gitta Kutyniok, and Giuseppe Caire. RadioUNet: Fast Radio Map Estimation with Convolutional Neural Networks. *IEEE Transactions on Wireless Communications*, 20(6):4001–4015, 2021.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

Haofan Lu, Christopher Vattheuer, Baharan Mirzasoleiman, and Omid Abari. NeWRF: A Deep Learning Framework for Wireless Radiation Field Reconstruction and Channel Prediction. In *Proceedings of the 41st International Conference on Machine Learning*, pages 33147–33159, 2024.

Ségolène Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. PnP-Flow: Plug-and-Play Image Restoration with Flow Matching. In *International Conference on Learning Representations*, 2025.

Jason D. McEwen and Yves Wiaux. Compressed Sensing for Radio Interferometric Imaging: Review and Future Direction. In *IEEE International Conference on Image Processing*, pages 1313–1316, 2011.

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 Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. FlowChef: Steering of Rectified Flow Models for Controlled Generations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15308–15318, 2025.

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

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample Flow Matching: Straightening Flows with Minibatch Couplings. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28100–28127, 2023.

Mehrsa Pourya, Bassam El Rawas, and Michael Unser. FLOWER: A Flow-Matching Solver for Inverse Problems. In *International Conference on Learning Representations*, 2026.

Saiprasad Ravishankar, Jong Chul Ye, and Jeffrey A. Fessler. Image Reconstruction: From Sparsity to Data-Adaptive Methods and Machine Learning. *Proceedings of the IEEE*, 108(1):86–109, 2020.

Nicholas Rawlinson, Andreas Fichtner, Malcolm Sambridge, and Mallory K. Young. Seismic Tomography and the Assessment of Uncertainty. *Advances in Geophysics*, 55:1–76, 2014.

François Roddier. Interferometric Imaging in Optical Astronomy. *Physics Reports*, 170(2):97–166, 1988.

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.

Daniel Romero and Seung-Jun Kim. Radio Map Estimation: A Data-Driven Approach to Spectrum Cartography. *IEEE Signal Processing Magazine*, 39(6):53–72, 2022.

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.

Rajesh Shrestha and Xiao Fu. Taming Score-Based Denoisers in ADMM: A Convergent Plug-and-Play Framework. In *International Conference on Learning Representations*, 2026.

Sagar Shrestha, Xiao Fu, and Mingyi Hong. Deep Spectrum Cartography: Completing Radio Map Tensors Using Learned Neural Models. *IEEE Transactions on Signal Processing*, 70:1170–1184, 2022.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-Guided Diffusion Models for Inverse Problems. In *International Conference on Learning Representations*, 2023.

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

Hao Sun, Shicong Liu, Xianghao Yu, and Ying Sun. Flow Matching-Based Active Learning for Radio Map Construction with Low-Altitude UAVs. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21406–21410, 2026.

Yves Teganya and Daniel Romero. Deep Completion Autoencoders for Radio Map Estimation. *IEEE Transactions on Wireless Communications*, 21(3):1710–1724, 2022.

Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. Plug-and-Play Priors for Model Based Reconstruction. In *IEEE Global Conference on Signal and Information Processing*, pages 945–948, 2013.

Jean Virieux and Stéphane Operto. An Overview of Full-Waveform Inversion in Exploration Geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model. In *International Conference on Learning Representations*, 2023.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Chaozheng Wen, Jingwen Tong, Yingdong Hu, Zehong Lin, and Jun Zhang. WRF-GS: Wireless Radiation Field Reconstruction with 3D Gaussian Splatting. In *IEEE INFOCOM 2025 – IEEE Conference on Computer Communications*, pages 1–10, 2025.

Le Xu, Lei Cheng, Junting Chen, Wenqiang Pu, and Xiao Fu. Radio Map Estimation via Latent Domain Plug-and-Play Denoising. *IEEE Transactions on Signal Processing*, 74:3240–3253, 2026.

Yong Zeng and Xiaoli Xu. Toward Environment-Aware 6G Communications via Channel Knowledge Map. *IEEE Wireless Communications*, 28(3):84–91, 2021.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 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 *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yifang Zhang and Shaowei Wang. K-Nearest Neighbors Gaussian Process Regression for Urban Radio Map Reconstruction. *IEEE Communications Letters*, 26(12):3049–3053, 2022.

Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. NeRF^2: Neural Radio-Frequency Radiance Fields. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 393–407, 2023.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1219–1229, 2023.

A PROOFS

Correction and error identities. Using $\mathbf{y} = \mathcal{A}_\Omega \operatorname{vec}(\mathbf{X}^\star) + \mathbf{n}$ and $\mathcal{B}_\lambda = \mathbf{I}_P - \mathcal{T}_\lambda \mathcal{A}_\Omega$, we obtain

$$\begin{aligned}\operatorname{vec}\Big(\widetilde{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\Big) &= (\mathcal{T}_\lambda \mathcal{A}_\Omega - \mathbf{I}_P)\operatorname{vec}(\mathbf{X}^\star) + \mathcal{B}_\lambda \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k}) + \mathcal{T}_\lambda \mathbf{n} \\ &= \mathcal{B}_\lambda \operatorname{vec}\Big(\widehat{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\Big) + \mathcal{T}_\lambda \mathbf{n},\end{aligned}$$

which is Equation (23).

For the complete-field-regularized estimator, the first-order optimality condition of Equation (20) is

$$\big(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\big) \operatorname{vec}(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k}) = \mathcal{A}_\Omega^\top \mathbf{y} + \gamma \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k}). \tag{27}$$

Since $\gamma > 0$ and both $\mathcal{A}_\Omega^\top \mathcal{A}_\Omega$ and $\mathbf{L}$ are positive semidefinite, the coefficient matrix is positive definite. Thus the unique minimizer is

$$\operatorname{vec}(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k}) = \big(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\big)^{-1} \Big[\mathcal{A}_\Omega^\top \mathbf{y} + \gamma \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k})\Big]. \tag{28}$$

Subtracting the coefficient matrix applied to $\operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k})$ from both sides of Equation (27) gives Equation (21). When $\mathbf{e}_{t_k} = \mathbf{0}$, the residual system has the unique solution $\widehat{\mathbf{R}}_{t_k} = \mathbf{0}$. The correction of the complete-field-regularized estimator vanishes if and only if $\mathbf{L} \operatorname{vec}(\widehat{\mathbf{X}}_{1,t_k}) = \mathbf{0}$, since $\lambda > 0$ and its coefficient matrix is invertible.

To obtain the complete-field error identity, subtract the same coefficient matrix applied to $\operatorname{vec}(\mathbf{X}^\star)$ from both sides of Equation (27) and use the observation model:

$$\begin{aligned}&\big(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\big) \operatorname{vec}\Big(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k} - \mathbf{X}^\star\Big) \\ &\quad = \gamma \operatorname{vec}\Big(\widehat{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\Big) - \lambda \mathbf{L} \operatorname{vec}(\mathbf{X}^\star) + \mathcal{A}_\Omega^\top \mathbf{n}.\end{aligned}$$

Multiplying by the inverse coefficient matrix yields Equation (24).

Proof of Proposition 5.1. The sampling matrix, spatial matrix, and regularization parameters are fixed. For the pretrained model, $\widehat{\mathbf{X}}_{1,t_k}$ is a deterministic function of $(\mathbf{X}_{t_k}, t_k)$ and can therefore be taken outside the corresponding conditional expectation. Taking conditional expectations in Equation (23) gives

$$\begin{aligned}&\mathbb{E}\Big[\operatorname{vec}\Big(\widetilde{\mathbf{X}}_{1,t_k} - \mathbf{X}^\star\Big) \mid \mathbf{X}_{t_k}, t_k\Big] \\ &\quad = \mathcal{B}_\lambda \operatorname{vec}\Big(\widehat{\mathbf{X}}_{1,t_k} - \boldsymbol{\mu}_{t_k}\Big) + \mathcal{T}_\lambda \mathbb{E}[\mathbf{n} \mid \mathbf{X}_{t_k}, t_k].\end{aligned}$$

Under $\widehat{\mathbf{X}}_{1,t_k} = \boldsymbol{\mu}_{t_k}$ and $\mathbb{E}[\mathbf{n} \mid \mathbf{X}_{t_k}, t_k] = \mathbf{0}$, both terms vanish. This proves Equation (25).

Similarly, Equation (24) gives

$$\begin{aligned}&\mathbb{E}\Big[\operatorname{vec}\Big(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k} - \mathbf{X}^\star\Big) \mid \mathbf{X}_{t_k}, t_k\Big] \\ &\quad = \big(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\big)^{-1} \Big[\gamma \operatorname{vec}\Big(\widehat{\mathbf{X}}_{1,t_k} - \boldsymbol{\mu}_{t_k}\Big) - \lambda \mathbf{L} \operatorname{vec}(\boldsymbol{\mu}_{t_k}) \\ &\qquad\qquad + \mathcal{A}_\Omega^\top \mathbb{E}[\mathbf{n} \mid \mathbf{X}_{t_k}, t_k]\Big].\end{aligned}$$

The conditional prediction-error term and the conditional noise term vanish under the same assumptions, leaving

$$\begin{aligned}&\mathbb{E}\Big[\operatorname{vec}\Big(\widehat{\mathbf{X}}^{\mathrm{cf}}_{1,t_k} - \mathbf{X}^\star\Big) \mid \mathbf{X}_{t_k}, t_k\Big] \\ &\quad = -\big(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L} + \gamma \mathbf{I}_P\big)^{-1} \lambda \mathbf{L} \operatorname{vec}(\boldsymbol{\mu}_{t_k}),\end{aligned}$$

which proves Equation (26). Because the coefficient matrix is invertible and $\lambda > 0$, this bias is zero if and only if $\mathbf{L} \operatorname{vec}(\boldsymbol{\mu}_{t_k}) = \mathbf{0}$. □

Table 3: Dataset composition and evaluation roles. All fields have shape 128×128. "Evaluated" denotes the number of fields used in the corresponding experiment. Tx and Rx denote transmitter and receiver, respectively.

Setting	Generated	Evaluated	Role
Etoile, 5 Tx, 25 m Tx / 60 m Rx	2,000	–	Prior training
Etoile, 5 Tx, 25 m Tx / 60 m Rx	100	100	Validation
Etoile, 5 Tx, 25 m Tx / 60 m Rx	100	100	In-distribution test
Etoile, 1 Tx, 25 m Tx / 60 m Rx	100	100	Deployment shift
Other deployment or scene shifts	100 each	100 each	Held-out transfer

B Implementation and Reproducibility Details

This appendix provides the implementation details needed to reproduce the experiments. Unless stated otherwise, all settings are fixed before test evaluation.

B.1 Data Generation and Data Splits

Ray-tracing configuration. For the reference Etoile setting described in Section 6.1, each deployment contains five independently positioned transmitters at a height of 25 m within a centered 500×400 m placement region. Transmitter positions and azimuths are sampled independently. Each transmitter radiates at 23 dBm through an 8×2 vertically polarized array specified in 3rd Generation Partnership Project (3GPP) Technical Report (TR) 38.901 (3GPP, 2026), with half-wavelength spacing, and the receiver uses a single isotropic vertically polarized element. Ray tracing uses a 100 MHz bandwidth, a maximum interaction depth of five, and 10^6 ray samples per transmitter. The field value at each grid location is the maximum received signal strength over the five transmitter links.

Data splits and shifted settings. The training, validation, and test sets contain disjoint transmitter deployments and no duplicate fields. The training set contains 2,000 fields. The 100 validation fields are used for hyperparameter selection, while the 100 test fields are used only for final evaluation.

For transfer evaluation, we additionally consider changes in transmitter count, transmitter height, receiver height, and urban scene. Each shifted setting contains 100 fields. None of the shifted settings is used for training or hyperparameter selection.

Table 3 summarizes the datasets used for training, validation, and evaluation.

B.2 Pretrained Flow Matching Prior

Architecture and objective. The prior is a single-channel, time-conditioned U-Net with base width 64, channel multipliers $(1, 2, 4)$, ConvNeXt blocks, linear attention at each resolution, bottleneck attention, and a full-resolution encoder–decoder skip. It contains 14,888,231 trainable parameters and predicts a single velocity channel. Training samples $\mathbf{X}_0 \sim \mathcal{N}(0, I)$ and $t \sim \mathcal{U}[0, 1]$, forms $\mathbf{X}_t = (1 - t)\mathbf{X}_0 + t\mathbf{X}_1$, and minimizes $\|\mathbf{v}_\theta(\mathbf{X}_t, t) - (\mathbf{X}_1 - \mathbf{X}_0)\|_F^2$. Only complete training fields are used; sparse measurements and inverse-problem losses are not included during training.

Optimization and checkpoint selection. We train with Adam, batch size 64, and a cosine learning-rate schedule from 10^{-4} to 10^{-6} over at most 50,000 updates. Training uses seed 42 and bfloat16 automatic mixed precision; validation uses a fixed independent seed (4,242) every 500 updates. The checkpoint at update 11,000 minimizes the validation velocity MSE and is used for all subsequent wireless-field experiments.

B.3 Normalization and Numerical Details

Normalization and measurement noise. The training-set extrema $a = -219.645$ dBm and $b = -45.151$ dBm are used to linearly scale each field to $[-1, 1]$ as $2(x - a)/(b - a) - 1$. Gaussian measurement noise is added in the physical dB domain before normalization, and all reported metrics

are computed after converting the reconstructed fields back to dBm. The main experiments use measurement noise with a standard deviation of 3 dB.

Spatial regularization. For the main baseline comparison in Section 6.3 and the additional wireless experiments in Appendix D, the neighborhood set $\mathcal{E}$ in Equations (12)–(13) is obtained by connecting locations whose Euclidean distance on a fixed 500×400 reference grid is no greater than $\rho = 4$. The corresponding spacings of the 128×128 grid are $500/128$ and $400/128$ along the two dimensions. Therefore, the radius includes the immediate horizontal and vertical pairs but excludes diagonal pairs, giving the four-neighbor structure used in the main experiments. For each pair $\{i, j\} \in \mathcal{E}$, we set

$$w_{ij} = \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}{2\sigma_s^2}\right), \tag{29}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are the reference-grid coordinates and σ_s controls the spatial decay. These pairwise weights define the spatial regularization matrix $\mathbf{L}$ in Equation (13). The reference grid is used only to parameterize the regularizer and does not represent the physical receiver-plane extent. We set $\sigma_s = 3$ at all four main sampling rates.

Linear solver. For noisy measurements, the residual system is solved in 32-bit floating-point (FP32) using Jacobi-preconditioned conjugate gradients. We use a relative residual tolerance of 2×10^{-4} and a maximum of 200 iterations. The coefficient matrix and preconditioner are prepared once for each reconstruction and reused at all Flow steps.

C Evaluation Protocol

The wireless experiments in Section 6 and Appendix D use validation-selected hyperparameters, fixed before testing. All accuracy results are averaged over 100 test fields per condition.

C.1 Evaluation Settings

Table 4 summarizes the sampling rates, noise levels, and deployment conditions for each comparison.

Table 4: Experimental settings used in the main and additional evaluations.

Experiment	Conditions
Sampling rate	$1, 2, 5, 10\%$; uniform sampling; 3 dB noise
Noise level	$2, 5\%$; clean, 1, 3, 5, and 8 dB; uniform sampling
Sampling geometry	$2, 5\%$; uniform, random flight, and crowdsourced; 3 dB noise
Residual vs. complete field	$1, 2, 5, 10\%$; uniform sampling; 3 dB noise
Component ablations	$1, 2, 5, 10\%$; uniform sampling; 3 dB noise
Generalization	10%; 3 dB noise; deployment and scene shifts

C.2 Paired Evaluation and Sampling Patterns

At a sampling rate r, we retain $\max\{1, \mathrm{round}(rP)\}$ observed locations. For the 128×128 fields used in our experiments, the sampling rates of 1%, 2%, 5%, and 10% correspond to 164, 328, 819, and 1,638 measurements, respectively. For each test field, the sampling masks are nested across sampling rates, so measurements used at a lower rate are retained when the sampling rate increases.

All methods are evaluated on the same test field and receive identical sampling locations and noisy measurements. Methods that start from a Gaussian source share the same initial realization. Flow-RI draws a new independent Gaussian source at every feedback step, as specified in Algorithm 1.

Sampling patterns. Uniform sampling selects locations randomly over the complete grid. The random-flight pattern follows a connected trajectory with one-cell moves and a maximum turning angle of 25°. Coverage-aware waypoints are used to encourage the trajectory to explore different parts of the field, and the 2% trajectory is taken as a prefix of the corresponding 5% trajectory.

For crowdsourced sampling, 70% of the observations are drawn from three to six spatially separated activity regions, while the remaining 30% are sampled uniformly over the field. Duplicate locations are resampled until the required number of unique observations is obtained. Thus, the three sampling patterns use the same number of measurements at a given sampling rate and differ only in their spatial distribution.

C.3 Hyperparameter Selection

Hyperparameters are selected on the validation set using the candidate values listed in Table 5. For each method, we select the configuration with the lowest validation MAE.

Table 5: Hyperparameters used in validation.

Method	Parameter	Description	Candidate values
KNN	K	Number of nearest neighbors	$\{1, 3, 5, 8\}$
DPS-ODE	N	Number of Flow Matching evaluations	$\{8, 12, 20, 30\}$
	η	Measurement-guidance strength	$\{4, 5, 6, 7, 8\}$
PnP-Flow	N	Number of Flow Matching evaluations	100 (fixed)
	α	Exponent in $\gamma_t = \mathrm{lr}(1-t)^\alpha$	$\{10^{-3}, 10^{-2}, 0.1, 0.3, 0.5, 1\}$
	lr	Scale in the data-step schedule γ_t	$\{1, 1.5, 2, 2.5, 3, 3.25, 3.5\}$
FlowDPS	N	Number of Flow Matching evaluations	100 (fixed)
	s	Consistency-update size	$\{10, 20, 40, 60\}$
FlowChef	N	Number of Flow Matching evaluations	100 (fixed)
	γ_{g}	Guidance strength	$\{3000, 4000, 4800, 6000\}$
FLOWER	N	Number of Flow Matching evaluations	100 (fixed)
	p	Grid-power parameter	$\{0.2, 0.25, 0.5, 1\}$
Flow-CR	N	Number of Flow Matching evaluations	$\{3, 4, 5\}$
	λ	Spatial regularization strength	$\{0.003, 0.01, 0.03, 0.1\}$
	γ	Reference anchoring strength	$\{10^k, 3\times10^k \mid k = -5, \ldots, 0\}$
Flow-RI	N	Number of Flow Matching evaluations	$\{3, 4, 5\}$
	λ	Spatial regularization strength	$\{0.003, 0.01, 0.03, 0.1\}$

For the 3 dB uniform-sampling experiments, the selected hyperparameters at different sampling rates are summarized in Table 6. For the comparison in Section 6.2, Flow-RI and Flow-CR use the unit-weight four-neighbor regularizer specified there. In Appendix C.4, Flow-RI, spatial-only reconstruction, and terminal-only correction use the same unit-weight regularizer. The remaining wireless Flow-RI experiments use the distance-weighted regularizer described in Appendix B.3. The selected parameters are then fixed for the corresponding test experiments, including the noise and sampling-geometry evaluations.

Table 6: Hyperparameters selected on the validation set for the 3 dB uniform-sampling experiment.

Method	Parameters	1%	2%	5%	10%
KNN	K	3	5	5	5
DPS-ODE	N/η	20/7	8/7	12/6	12/6
PnP-Flow	α/lr	0.1/3.5	0.1/3.5	0.01/3	0.001/2.5
FlowDPS	s	40	40	40	60
FlowChef	γ_{g}	4800	4800	4000	3000
FLOWER	p	0.2	0.2	0.2	0.25
Flow-CR	$N/\lambda/\gamma$	$5/0.003/10^{-5}$	$5/0.003/10^{-5}$	5/0.01/0.0003	5/0.01/0.0003
Flow-RI	N/λ	5/0.01	5/0.01	5/0.01	5/0.03

C.4 Component Ablations

We consider three component ablations of Flow-RI: (i) *spatial-only reconstruction*, which removes the Flow Matching prior; (ii) *pointwise residual correction*, which retains the Flow Matching prediction

and iterative feedback but removes spatial coupling from the residual estimation; and (iii) *terminal-only correction*, which retains the Flow Matching prior and spatial residual estimator but applies the residual correction only once after the final prediction. All comparisons use the same 3 dB noise setting and paired inputs as Section 6.2.

Spatial-only reconstruction uses only the measurements and the unit-weight four-neighbor spatial regularizer. It reconstructs the field by solving

$$(\mathcal{A}_\Omega^\top \mathcal{A}_\Omega + \lambda \mathbf{L})\,\mathrm{vec}(\widehat{\mathbf{X}}) = \mathcal{A}_\Omega^\top \mathbf{y}.$$

No Flow Matching prediction is used in this variant.

Pointwise residual correction uses the same frozen Flow Matching model, five Flow Matching evaluations, and feedback rule as Flow-RI, but removes spatial coupling from the residual estimation. At each flow step, it solves

$$\widehat{\mathbf{R}}_t = \arg\min_{\mathbf{R}} \frac{1}{2}\,\|\mathcal{A}_\Omega\,\mathrm{vec}(\mathbf{R}) - \mathbf{e}_t\|_2^2 + \frac{\alpha}{2}\,\|\mathrm{vec}(\mathbf{R})\|_2^2\,.$$

This objective does not couple different spatial locations, so residual corrections are confined to the observed locations. When $\alpha = 0$, we use the minimum-norm solution, which sets the residual to zero at unobserved locations.

Terminal-only correction keeps the same five Flow Matching evaluations and Gaussian source sequence as Flow-RI, but does not apply residual correction at the intermediate flow steps. The intermediate states are formed from the uncorrected Flow Matching predictions, and the spatial residual solve in Equation (14) is applied once to the final endpoint prediction.

The spatial-only search uses $\lambda \in \{10^{-4}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}, 10^{-2}, 3\times10^{-2}, 0.1, 0.3, 1, 3\}$; the pointwise search uses $\alpha \in \{0, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3\}$. Minimizing the validation criterion selects $\lambda = 0.003$ for spatial-only reconstruction and $\alpha = 0$ for pointwise residual correction at all four sampling rates.

To isolate the effect of intermediate feedback, terminal-only correction uses the same λ as Flow-RI at each sampling rate; the two methods therefore differ only in when the residual correction is applied.

Table 7: Component ablations under uniform sampling with 3 dB measurement noise. Flow-RI, spatial-only reconstruction, and terminal-only correction use the unit-weight four-neighbor spatial regularizer where applicable.

Method	1%	2%	5%	10%
Flow-RI	**6.018**	**5.092**	**4.222**	**3.676**
Spatial only	9.443	6.735	4.758	3.887
Pointwise residual	22.551	21.621	19.044	15.351
Terminal-only correction	7.998	6.345	4.819	4.011

The three ablations separate the roles of the learned prior, spatial residual propagation, and intermediate feedback (Table 7). The gap between spatial-only reconstruction and Flow-RI is much larger at 1% than at 10% sampling, consistent with the learned prior being most useful when measurements are scarce. Removing spatial coupling from the residual estimator substantially increases MAE, indicating that pointwise residual correction alone is insufficient in this setting. The terminal-only result isolates the timing of the correction: using the same spatial residual correction only at the endpoint is consistently worse, so the benefit comes not only from the final correction but also from feeding it back into subsequent flow states.

C.5 Evaluation Metrics

We evaluate reconstruction performance using MAE, NMSE, PSNR, and SSIM. All metrics are computed over the complete $H \times W$ field after converting the reconstructed values back to dBm. Predictions are not clipped, and finite outliers are included in the averages. NMSE is a dimensionless ratio computed from the dBm-valued fields.

The MAE is defined as

$$\mathrm{MAE} = \frac{1}{P}\left\|\mathrm{vec}(\widehat{\mathbf{X}} - \mathbf{X}^\star)\right\|_1, \tag{30}$$

where $\widehat{\mathbf{X}}$ and $\mathbf{X}^\star$ denote the reconstructed and ground-truth fields, respectively. The NMSE is

$$\mathrm{NMSE} = \frac{\left\|\widehat{\mathbf{X}} - \mathbf{X}^\star\right\|_F^2}{\left\|\mathbf{X}^\star\right\|_F^2}. \tag{31}$$

For PSNR, we use the fixed training-set range $b - a$:

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{(b-a)^2}{\mathrm{MSE}}\right), \qquad \mathrm{MSE} = \frac{1}{P}\left\|\widehat{\mathbf{X}} - \mathbf{X}^\star\right\|_F^2. \tag{32}$$

SSIM is computed with an 11×11 Gaussian window with standard deviation 1.5 and reflection padding.

D Additional Experiments

D.1 Qualitative Comparison Across Sampling Rates

Figure 2 shows the same test field as Figure 1. At 1% sampling, the differences among methods are pronounced, with KNN exhibiting clear piecewise artifacts. At 10% sampling, the reconstructions are more similar across methods. The values shown below the panels are MAEs for this field rather than the test-set means in Table 2.

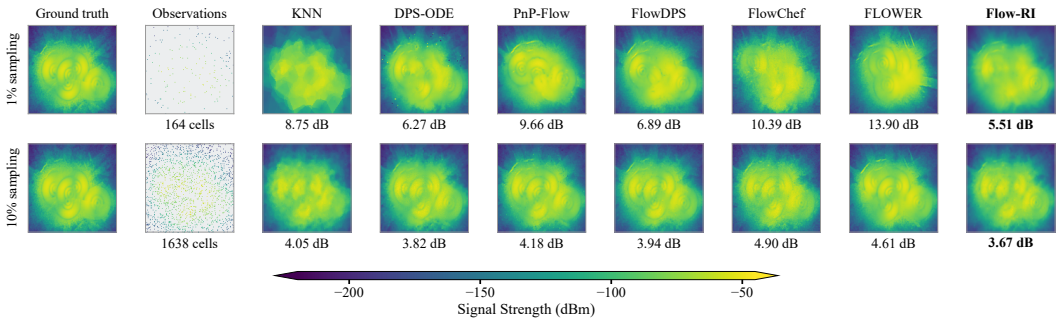


Figure 2: Qualitative reconstruction comparison at 1% (top) and 10% (bottom) uniform sampling with 3 dB measurement noise. Columns show the ground-truth field, sparse observations, six baselines, and Flow-RI.

D.2 Sensitivity to Noise-Level Mismatch

We test sensitivity to an incorrect noise assumption by fixing all hyperparameters at their 3 dB validation settings and varying the actual measurement noise. The noise standard deviation is 0 (clean), 1, 3, 5, or 8 dB, at sampling rates of 2% and 5%. Methods that use a noise estimate continue to assume 3 dB.

At 5% sampling, Flow-RI increases from 3.903 dB MAE in the clean case to 5.329 dB at 8 dB noise (Table 8). Flow-RI remains the lowest-MAE method throughout the sweep, but its margin over FlowDPS narrows at the highest noise level (5.329 versus 5.456 dB).

D.3 Robustness to Sampling Geometry

We compare uniform, random-flight, and crowdsourced sampling at 2% and 5% observation rates; the sampling patterns are defined in Appendix C.2. Each method retains its parameters selected for uniform sampling, and all methods receive the same fields, masks, and noise within each setting.

The large error under random-flight sampling is mainly a coverage effect rather than a change in measurement count (Table 9). At 2% sampling, Flow-RI increases from 5.094 dB under uniform sampling to 12.807 dB along the flight trajectory, while crowdsourced sampling remains at 5.499 dB.

Table 8: MAE (dB) under noise mismatch over 100 test fields. Noise denotes the actual standard deviation. All hyperparameters are fixed to the 3 dB settings.

Noise	KNN	DPS-ODE	PnP-Flow	FlowDPS	FlowChef	FLOWER	Flow-RI
2% sampling							
clean	6.369	6.093	7.369	5.841	8.569	11.278	**4.865**
1 dB	6.402	6.116	7.378	5.865	8.595	11.278	**4.899**
3 dB	6.560	6.304	7.452	5.975	8.731	11.304	**5.094**
5 dB	6.825	6.622	7.559	6.151	8.944	11.351	**5.419**
8 dB	7.377	7.200	7.791	6.561	9.367	11.457	**6.089**
5% sampling							
clean	4.649	4.485	5.105	4.506	5.919	6.166	**3.903**
1 dB	4.711	4.517	5.127	4.555	5.967	6.174	**3.961**
3 dB	4.945	4.737	5.265	4.745	6.202	6.237	**4.213**
5 dB	5.296	5.071	5.481	5.009	6.536	6.343	**4.593**
8 dB	5.983	5.645	5.883	5.456	7.114	6.564	**5.329**

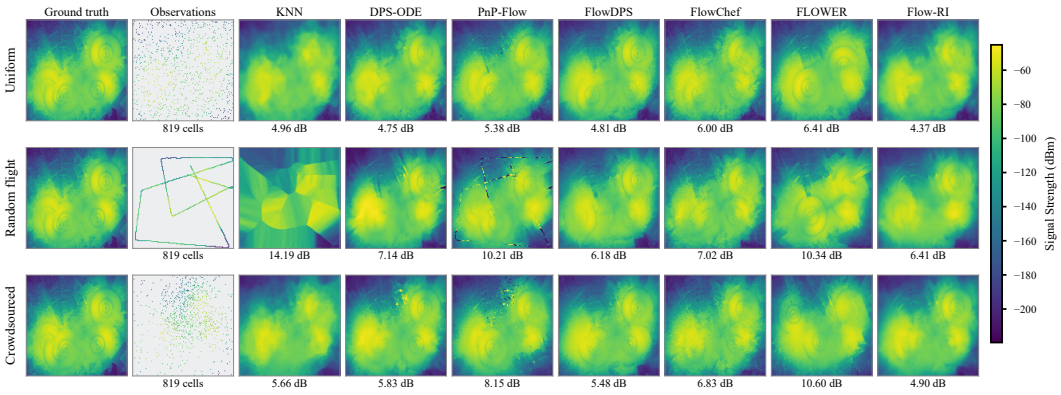


Figure 3: Sampling geometries at 5% sampling and 3 dB measurement noise. All rows use the same test field. Columns show the ground truth, observations, six baselines, and Flow-RI. Values below reconstructions are MAE (dB).

Table 9: MAE (dB) across sampling geometries on 100 test fields with 3 dB measurement noise. All methods retain their uniform-sampling parameters. Bold indicates the lowest mean in each row.

Geometry	KNN	DPS-ODE	PnP-Flow	FlowDPS	FlowChef	FLOWER	Flow-RI
2% sampling							
Uniform	6.560	6.304	7.452	5.975	8.731	11.304	**5.094**
Random flight	26.934	19.727	15.966	16.442	17.020	17.701	**12.807**
Crowdsourced	8.269	7.888	8.640	6.510	10.426	12.806	**5.499**
5% sampling							
Uniform	4.945	4.737	5.265	4.745	6.202	6.237	**4.213**
Random flight	14.061	10.383	10.438	7.871	9.446	10.860	**7.175**
Crowdsourced	5.740	5.842	6.084	5.079	7.059	7.709	**4.521**

The same ordering holds at 5%. Figure 3 makes the difference visible: the connected flight trajectory leaves broad regions without nearby observations, and the reconstructions lose spatial detail there. Flow-RI still gives the lowest MAE in all six settings, but this experiment exposes a clear limitation when spatial coverage is poor.

D.4 Cross-Scene and Deployment Generalization

We evaluate transfer at 10% sampling with 3 dB measurement noise. The reference setting is Etoile with five transmitters at a height of 25 m and a receiver plane at 60 m. We vary one deployment

attribute at a time: the transmitter count becomes one, three, or seven; the transmitter height increases to 35 m; or the receiver height decreases to 30 m. We also evaluate the five-transmitter setting in Florence, Munich, and San Francisco.

We use the checkpoint, normalization, observation protocol, Gaussian-source sampling procedure, and inference parameters from the 10% in-distribution experiment for all transfer settings. For each unseen city, we center the receiver plane on that scene while retaining Etoile's physical field of view and 128×128 resolution. Each setting contains 100 test fields, evaluated with protocol seed 0. None of the shifted datasets is used for training or parameter selection.

Flow-RI has the lowest MAE in the reference setting and all eight transfer settings (Table 10). Changes in transmitter count give MAEs of 3.480–3.855 dB, close to the reference error of 3.657 dB. Height changes are more difficult, particularly when the receiver plane is lowered to 30 m, where MAE reaches 5.807 dB.

On the three unseen cities, KNN is the strongest baseline. Compared with KNN, Flow-RI reduces MAE from 4.330 to 4.031 dB in Florence, from 5.014 to 4.691 dB in Munich, and from 5.718 to 5.444 dB in San Francisco. The learned prior is not adapted for any of the three cities.

Figure 4 shows one reconstruction per unseen city. Flow-RI recovers the main high-signal regions, although fine spatial structure is lost. Among these three cities, San Francisco has the highest MAE.

Table 10: MAE (dB) over 100 test fields per scene or deployment, with 10% sampling and 3 dB measurement noise. Parameters selected for the in-distribution (ID) setting are used throughout. Bold indicates the lowest mean in each row.

Scene and deployment	KNN	DPS-ODE	PnP-Flow	FlowDPS	FlowChef	FLOWER	Flow-RI
Etoile / 5 Tx / Tx 25 m / Rx 60 m (ID)	4.110	3.958	4.227	4.042	5.233	4.586	**3.657**
Etoile / 1 Tx / Tx 25 m / Rx 60 m	3.624	4.637	7.172	6.167	11.973	9.104	**3.559**
Etoile / 3 Tx / Tx 25 m / Rx 60 m	4.299	4.177	4.869	4.567	6.044	5.596	**3.855**
Etoile / 7 Tx / Tx 25 m / Rx 60 m	3.853	4.004	4.050	3.858	5.043	4.408	**3.480**
Etoile / 5 Tx / Tx 35 m / Rx 60 m	4.772	5.186	5.447	5.216	6.533	5.749	**4.464**
Etoile / 5 Tx / Tx 25 m / Rx 30 m	6.162	7.001	8.345	7.727	10.896	9.045	**5.807**
Florence / 5 Tx / Tx 25 m / Rx 60 m	4.330	5.023	4.995	4.755	6.101	5.314	**4.031**
Munich / 5 Tx / Tx 25 m / Rx 60 m	5.014	5.421	5.835	5.536	7.105	6.119	**4.691**
San Francisco / 5 Tx / Tx 25 m / Rx 60 m	5.718	6.787	8.260	7.599	11.320	9.160	**5.444**

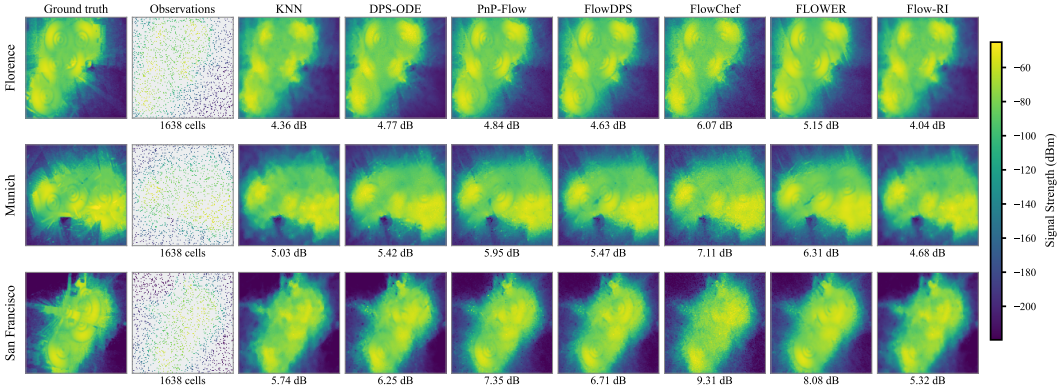


Figure 4: Reconstructions in three unseen cities with five transmitters, 10% sampling, and 3 dB measurement noise.

D.5 Runtime and Computational Cost

Timing protocol. We measure reconstruction time after loading the model and input data. Each invocation includes method-specific setup. For Flow-RI, this covers spatial-system construction, solver preparation, Gaussian source sampling, flow evaluations, and residual solves. The spatial system is reused across steps. Each condition uses 20 warmup runs followed by 100 timed repetitions, with CUDA synchronization before and after each invocation. The hardware and software are an RTX 5070 Laptop GPU (8,150.6 MB), CUDA 12.8, Python 3.10.18, and PyTorch 2.8.0.dev20250615+cu128.

Timing results. Table 11 reports the inference time and number of function evaluations (NFE). KNN is the fastest method, taking 0.0472 s and 0.0545 s per field at 2% and 5% sampling. Flow-RI takes 0.3350 s and 0.2418 s with five flow evaluations, compared with 1.1526 s and 1.7591 s for DPS-ODE with eight and twelve evaluations. The 100-NFE methods take 6.5209–7.8157 s per field.

Table 11: Mean per-field inference time and number of function evaluations (NFE) at batch size one, averaged over 100 timed repetitions.

Method	2% (s)	5% (s)	NFE (2%/5%)
KNN	0.0472	0.0545	0/0
DPS-ODE	1.1526	1.7591	8/12
PnP-Flow	6.5431	6.6153	100/100
FlowDPS	7.7626	7.8157	100/100
FlowChef	6.5209	6.5666	100/100
FLOWER	7.0250	7.0895	100/100
Flow-RI	0.3350	0.2418	5/5

E Natural Image Restoration

We also apply Flow-RI to random inpainting, denoising, box inpainting, and super-resolution with a pretrained natural-image Flow Matching prior. All methods use the same observations.

E.1 Experimental Setup

Data and pretrained prior. We use eight CelebA images for validation and eight images for testing (Liu et al., 2015).[1] The validation images are used only for parameter selection, and all reported results use the test split. All images are center-cropped to 178×178, bilinearly resized to 128×128, and scaled to $[-1, 1]$. All methods use the public CelebA optimal-transport Flow Matching U-Net from PnP-Flow (Martin et al., 2025), without fine-tuning. [2] The same measurements and Gaussian source samples are used across methods.

Restoration tasks and settings. Table 12 summarizes the four restoration tasks, their observation operators, noise levels, and Flow-RI parameters. Random inpainting uses a Bernoulli mask shared across RGB channels, while box inpainting removes a centered 40×40 region. For $2\times$ super-resolution, the observation operator retains the top-left pixel of each 2×2 block, producing a 64×64 observation without additional blurring.

Table 12: Natural-image restoration settings and Flow-RI parameters. Noise levels are defined on $[-1, 1]$.

Task	Observation operator	σ	λ
Random inpainting	30% observed pixels	0.01	0.003
Denoising	Identity	0.20	10
Box inpainting	Central 40×40 hole	0.05	1
$2\times$ super-resolution	Stride-two decimation	0.05	1

Image inference. We use the same Flow-RI reconstruction procedure as in Algorithm 1. The residual is estimated independently for each RGB channel using the same four-neighbor spatial regularization matrix $\mathbf{L}$. For all four tasks, Flow-RI uses $N = 100$ network evaluations on a uniform time grid. The coefficient matrix is fixed during reconstruction, so the solver components are prepared once and reused across flow steps.

[1] Validation images: 162771–162778; test images: 182638–182645.

[2] `https://github.com/annegnx/PnP-Flow`, commit `ee5b159`.

Parameter selection. For denoising, box inpainting, and super-resolution, we select λ from $\{0.003, 0.03, 0.3, 1, 3, 10, 30\}$ using mean PSNR on the validation images, with $N = 100$ fixed. Random inpainting uses $\lambda = 0.003$. The selected parameters are fixed for test evaluation.

Baselines and evaluation. We compare Flow-RI with PnP-Flow (Martin et al., 2025), FlowDPS (Kim et al., 2025), FlowChef (Patel et al., 2025), FLOWER (Pourya et al., 2026), Flow-Priors (Zhang et al., 2024), and optimal-transport ordinary differential equation (OT-ODE) (Pokle et al., 2024). PnP-Flow, Flow-Priors, and OT-ODE use the task-specific settings provided in the PnP-Flow implementation. For FlowDPS and FlowChef, we use their CelebA Flow Matching implementations and retain the random-inpainting settings for the other tasks.

The forward NFE budgets are 500 for PnP-Flow, 400 for FlowDPS, 300 for Flow-Priors, and 100 for FLOWER and Flow-RI. FlowChef uses at most 100 evaluations, while OT-ODE uses 140 for denoising and 180 for the other tasks. These budgets follow the corresponding implementations and are not matched across methods.

We report PSNR, SSIM (Wang et al., 2004), and AlexNet Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018). Reconstructions are clipped to $[-1, 1]$ before evaluation. PSNR is computed on $[0, 1]$ images in float64, and SSIM uses an 11×11 Gaussian window with standard deviation 1.5. All metrics are computed over the complete image, and the tables below report averages over the eight test images.

E.2 Results

The results are mixed across metrics. Flow-RI achieves the highest mean PSNR on three of the four tasks: random inpainting, box inpainting, and $2\times$ super-resolution, with gains of 0.737, 0.137, and 0.256 dB over the best baseline, respectively. The improvement is less consistent for SSIM and LPIPS. On random inpainting, Flow-RI also achieves the highest SSIM, although FlowDPS gives a lower LPIPS. On box inpainting, FLOWER performs better in both SSIM and LPIPS, while on super-resolution FLOWER has higher SSIM and OT-ODE has lower LPIPS.

Table 13: Random inpainting and denoising on eight CelebA-128 test images. Entries are averages over the eight images; bold indicates the best value in each column.

	Random inpainting (30% observed)			Denoising ($\sigma = 0.2$)		
Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
PnP-Flow	34.856	0.9599	0.0202	**32.549**	0.9111	0.0548
FlowDPS	34.530	0.9593	**0.0119**	27.012	0.6687	0.1871
FlowChef	29.203	0.8658	0.0626	20.809	0.3821	0.4197
FLOWER	33.926	0.9518	0.0165	32.497	**0.9139**	**0.0328**
Flow-Priors	33.805	0.9542	0.0168	29.705	0.7980	0.1466
OT-ODE	29.210	0.8788	0.0522	30.689	0.8619	0.0346
Flow-RI	**35.593**	**0.9651**	0.0130	31.042	0.8700	0.0872

Table 14: Box inpainting and $2\times$ super-resolution on the same eight CelebA-128 test images. Bold indicates the best value in each column. A dash indicates numerical failure of FlowChef on super-resolution.

	Box inpainting (40×40)			$2\times$ super-resolution		
Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
PnP-Flow	32.090	0.9418	0.0348	32.104	0.9098	0.0560
FlowDPS	31.367	0.9256	0.0237	33.181	0.9196	0.0184
FlowChef	18.393	0.7300	0.2116	—	—	—
FLOWER	32.712	**0.9483**	**0.0168**	33.083	**0.9287**	0.0323
Flow-Priors	31.281	0.8768	0.0391	28.939	0.7559	0.0979
OT-ODE	30.859	0.9199	0.0317	32.578	0.9221	**0.0183**
Flow-RI	**32.849**	0.9352	0.0209	**33.437**	0.9262	0.0409

Denoising is the main exception. Flow-RI obtains 31.042 dB PSNR, 0.8700 SSIM, and 0.0872 LPIPS, while both PnP-Flow and FLOWER perform better on all three metrics. FlowChef becomes numerically unstable on all eight super-resolution images after 80 or 81 evaluations; these failures are marked with dashes in Table 14.

To check sensitivity to the feedback sources, we repeat Flow-RI with three independently generated Gaussian source sequences while keeping the images, observations, initial states, and parameters fixed. Across the three runs, the sample standard deviation of the eight-image mean PSNR ranges from 0.023 to 0.197 dB (Table 15).

Table 15: Flow-RI performance across three Gaussian source sequences. Entries report the mean and sample standard deviation of the corresponding eight-image averages.

Task	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Random inpainting	35.667 ± 0.077	0.96547 ± 0.00037	0.01297 ± 0.00016
Denoising	31.065 ± 0.023	0.87059 ± 0.00060	0.08687 ± 0.00103
Box inpainting	32.862 ± 0.197	0.93492 ± 0.00076	0.02097 ± 0.00005
$2\times$ super-resolution	33.425 ± 0.035	0.92635 ± 0.00044	0.04077 ± 0.00013

Flow-RI improves PSNR on three of the four restoration tasks, but the eight-image test set, unmatched NFE budgets, and reuse of some baseline settings across tasks limit broader conclusions about natural-image restoration.

E.3 Qualitative Comparisons

Figures 5–8 show the reconstructions for all eight test images in filename order. Per-image PSNR, SSIM, and LPIPS are shown below each reconstruction.

CelebA-128: Denoising ($\sigma = 0.2$)

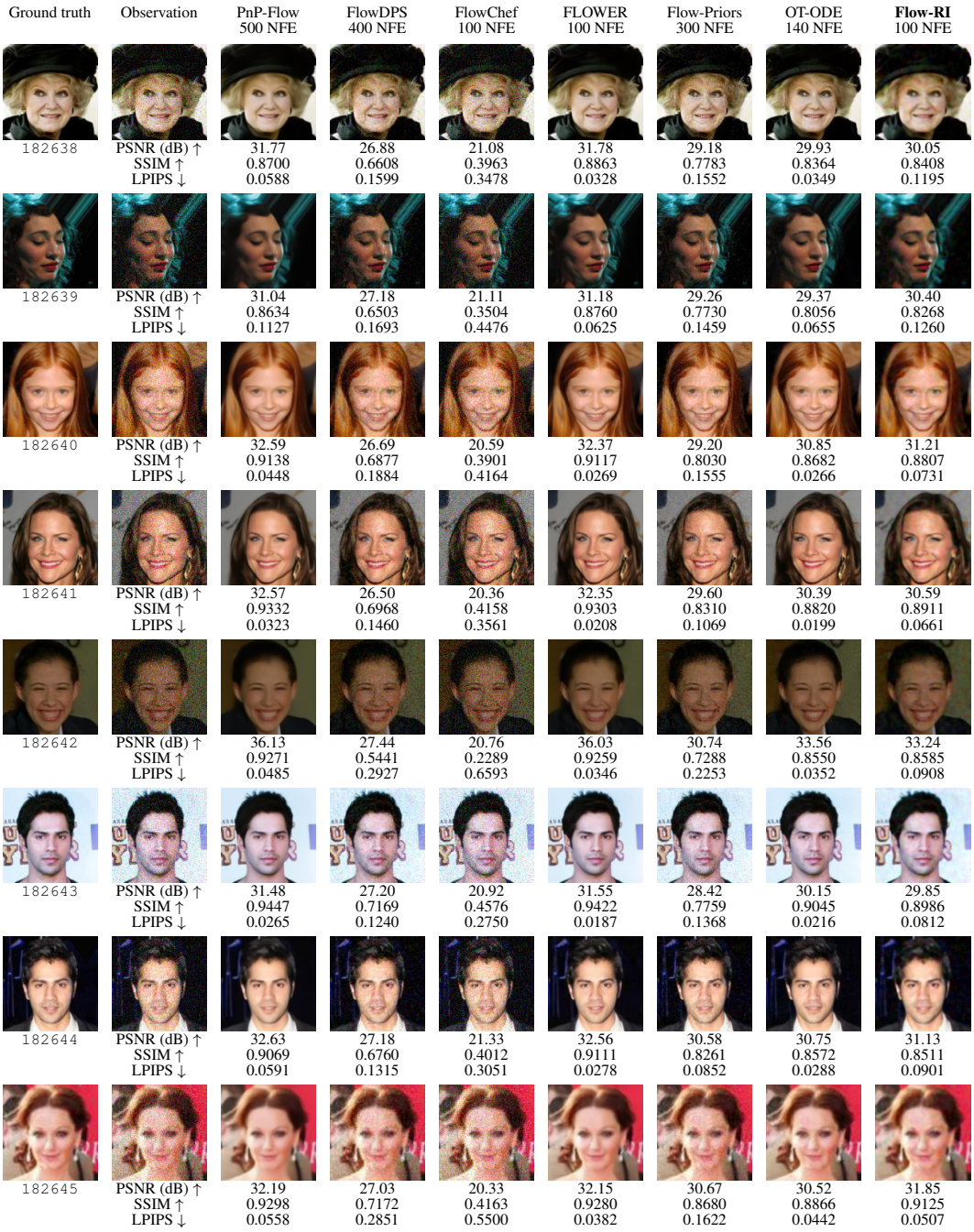


Figure 5: Denoising ($\sigma = 0.2$) on all eight CelebA-128 test images. Columns show the observation and reconstructions from each method. Per-image PSNR, SSIM, and LPIPS are reported below the corresponding reconstructions; Table 13 reports their averages. Headers give the forward NFE used by each method.

CelebA-128: $2\times$ super-resolution ($\sigma = 0.05$)

Ground truth	Observation	PnP-Flow 500 NFE	FlowDPS 400 NFE	FlowChef 100 NFE	FLOWER 100 NFE	Flow-Priors 300 NFE	OT-ODE 180 NFE	**Flow-RI** 100 NFE
								
182638	PSNR (dB) ↑	30.67	31.89	—	31.61	28.63	30.66	31.68
	SSIM ↑	0.8535	0.9104	—	0.9031	0.7409	0.9071	0.9083
	LPIPS ↓	0.0643	0.0179	—	0.0351	0.0677	0.0188	0.0611
								
182639	PSNR (dB) ↑	30.87	32.19	—	31.75	28.72	30.79	32.44
	SSIM ↑	0.8683	0.8930	—	0.8928	0.7352	0.8824	0.8935
	LPIPS ↓	0.1013	0.0241	—	0.0578	0.0786	0.0289	0.0651
								
182640	PSNR (dB) ↑	32.79	34.04	—	33.88	29.25	33.68	34.49
	SSIM ↑	0.9185	0.9291	—	0.9318	0.7883	0.9272	0.9367
	LPIPS ↓	0.0462	0.0107	—	0.0229	0.0771	0.0139	0.0277
								
182641	PSNR (dB) ↑	31.70	32.31	—	32.44	28.31	31.54	32.35
	SSIM ↑	0.9286	0.9256	—	0.9400	0.7792	0.9268	0.9327
	LPIPS ↓	0.0338	0.0146	—	0.0215	0.0930	0.0139	0.0304
								
182642	PSNR (dB) ↑	36.60	36.39	—	37.41	29.53	37.17	36.59
	SSIM ↑	0.9314	0.9101	—	0.9436	0.6506	0.9369	0.9254
	LPIPS ↓	0.0481	0.0247	—	0.0295	0.2058	0.0183	0.0403
								
182643	PSNR (dB) ↑	29.75	30.71	—	30.35	28.30	29.84	31.19
	SSIM ↑	0.9384	0.9321	—	0.9424	0.7861	0.9330	0.9397
	LPIPS ↓	0.0436	0.0219	—	0.0297	0.0647	0.0187	0.0450
								
182644	PSNR (dB) ↑	32.27	33.48	—	33.37	29.63	32.75	33.62
	SSIM ↑	0.9031	0.9149	—	0.9267	0.7731	0.9157	0.9186
	LPIPS ↓	0.0519	0.0177	—	0.0287	0.0497	0.0149	0.0368
								
182645	PSNR (dB) ↑	32.19	34.44	—	33.84	29.15	34.21	35.14
	SSIM ↑	0.9363	0.9419	—	0.9489	0.7938	0.9479	0.9545
	LPIPS ↓	0.0593	0.0152	—	0.0331	0.1469	0.0189	0.0209

Figure 6: $2\times$ super-resolution ($\sigma = 0.05$) on all eight CelebA-128 test images. Per-image PSNR, SSIM, and LPIPS are reported below the corresponding reconstructions; Table 14 reports their averages. Headers give the forward NFE used by each method. The observations are enlarged using nearest-neighbor interpolation. Gray cells indicate numerical failure of FlowChef, with the number of completed evaluations shown in each cell.

CelebA-128: Random inpainting (30% **observed,** $\sigma = 0.01$)

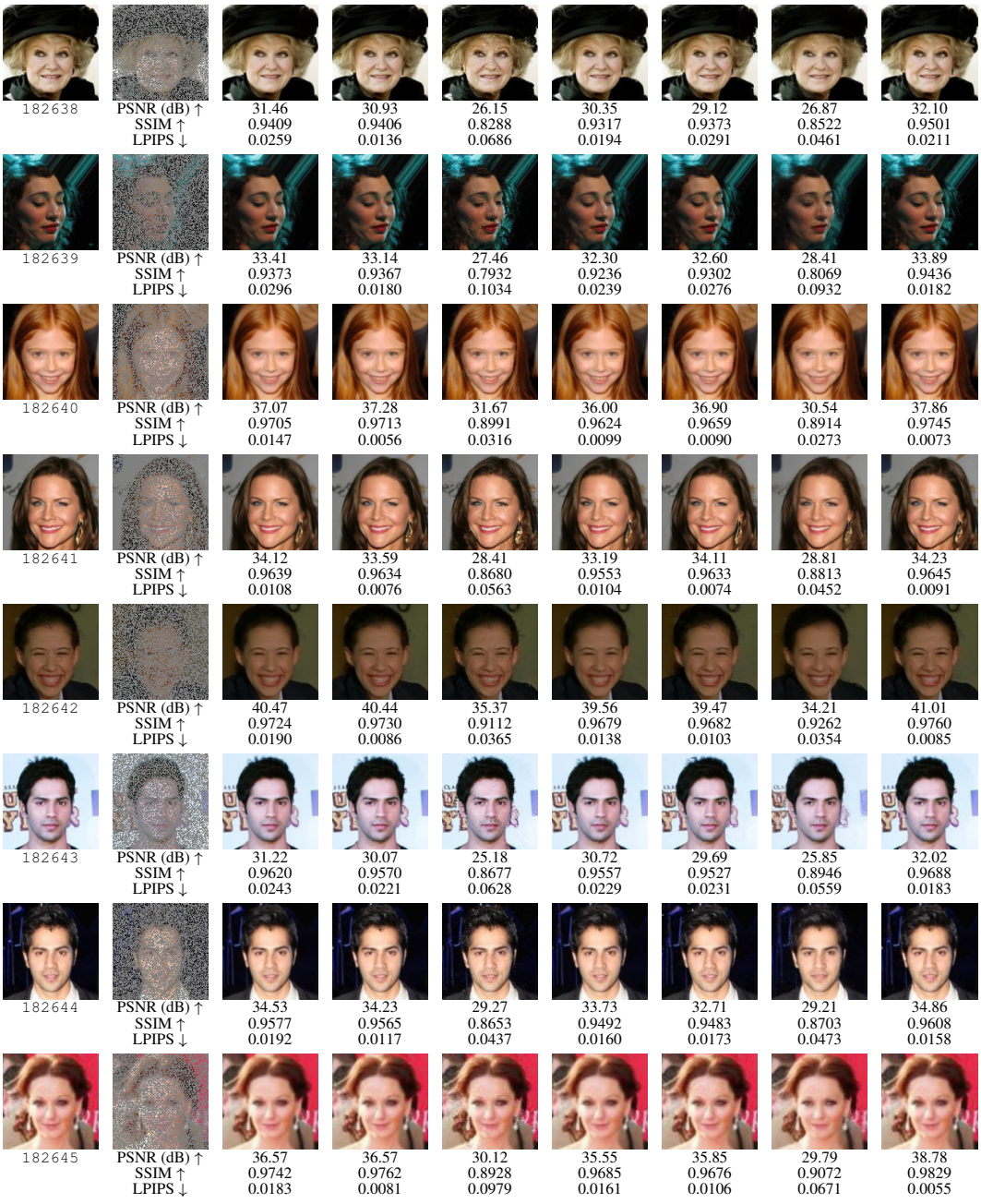


Figure 7: Random inpainting with 30% observed pixels and $\sigma = 0.01$ on all eight CelebA-128 test images. Per-image PSNR, SSIM, and LPIPS are reported below the corresponding reconstructions; Table 13 reports their averages. Headers give the forward NFE used by each method.

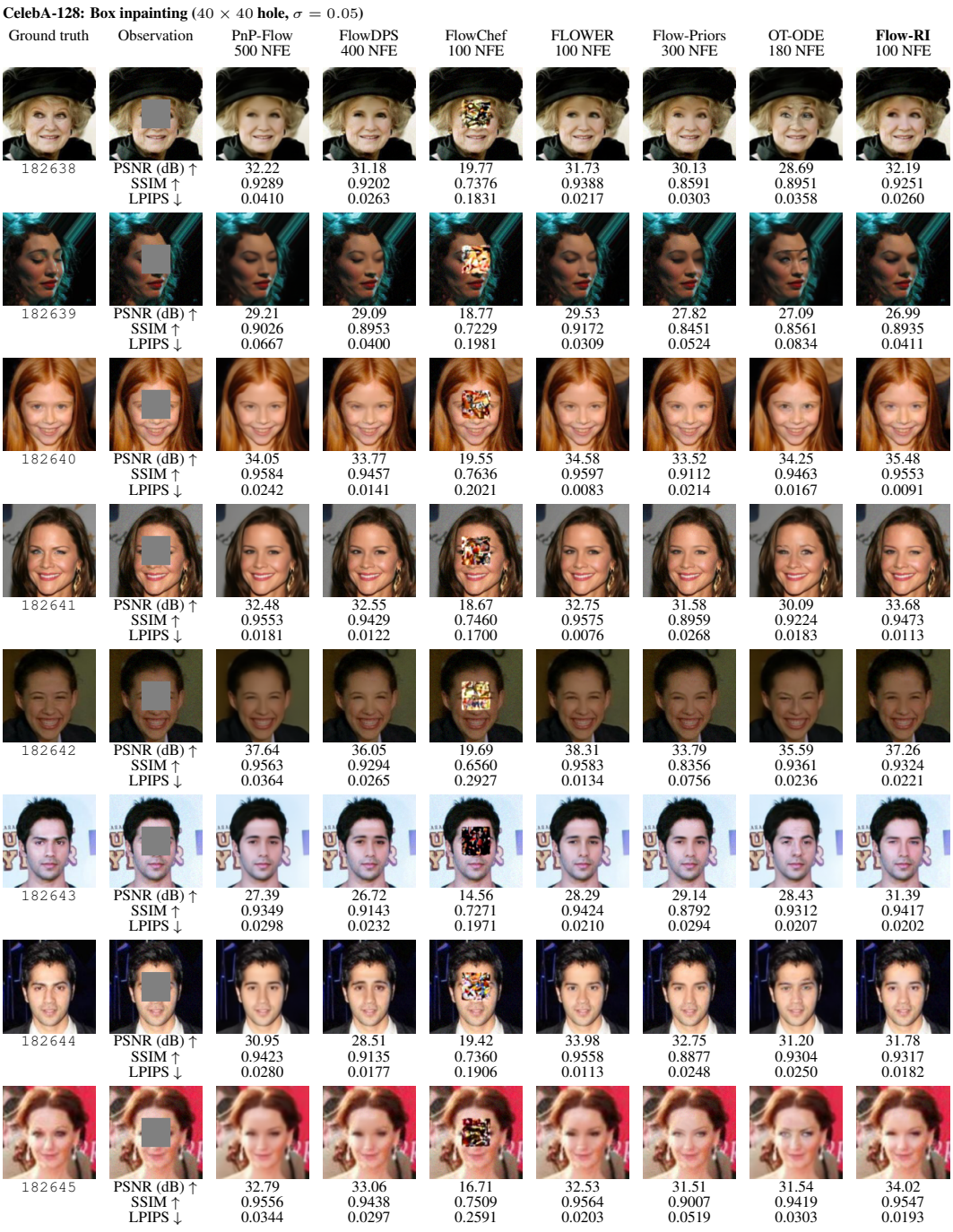


Figure 8: Box inpainting with a central 40×40 missing region and $\sigma = 0.05$ on all eight CelebA-128 test images. Per-image PSNR, SSIM, and LPIPS are reported below the corresponding reconstructions; Table 14 reports their averages. Headers give the forward NFE used by each method.