BAM! BAYESIAN ANYTHING MODEL: A FOUNDATION MODEL FOR GENERATIVE COMPUTATIONAL IMAGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models are transforming Bayesian computational imaging, yet the field still lacks physics-aware foundation models. Current practice falls into two camps. Large foundation image models are deployed as plug-and-play priors with zero-shot approximate likelihood guidance, which introduces significant bias and computational cost. Physics-aware generative models avoid this bias, but each is tied to a specific dataset, task and instrument. We introduce BAM (Bayesian Anything Model), a lightweight foundation model for few-step, physics-aware posterior sampling that generalises robustly to unseen data and tasks, zero-shot or with minimal finetuning. BAM upgrades the operator-conditioned Reconstruct Anything Model (RAM) backbone (Terris et al., 2026) into a conditional flow map, so instrument physics is specified at inference time rather than fixed during training. BAM has just 36M parameters and is pre-trained jointly on large image corpora and libraries of forward operators. A single network then draws posterior samples in a few steps, with no likelihood approximation and no guidance weights to tune. Across linear inverse problems on FFHQ, AFHQ, LSUN, DIV2K and the Köhler camera-shake benchmark, BAM outperforms in just 3 steps both specialised models and leading zero-shot methods in sample quality, at a fraction of their computational cost. BAM gives the community an accessible entry point to generative computational imaging, lowers the economic and environmental cost of training imaging models, and opens a new path for research on physics-aware Bayesian computational imaging. Code and weights will be released upon acceptance.

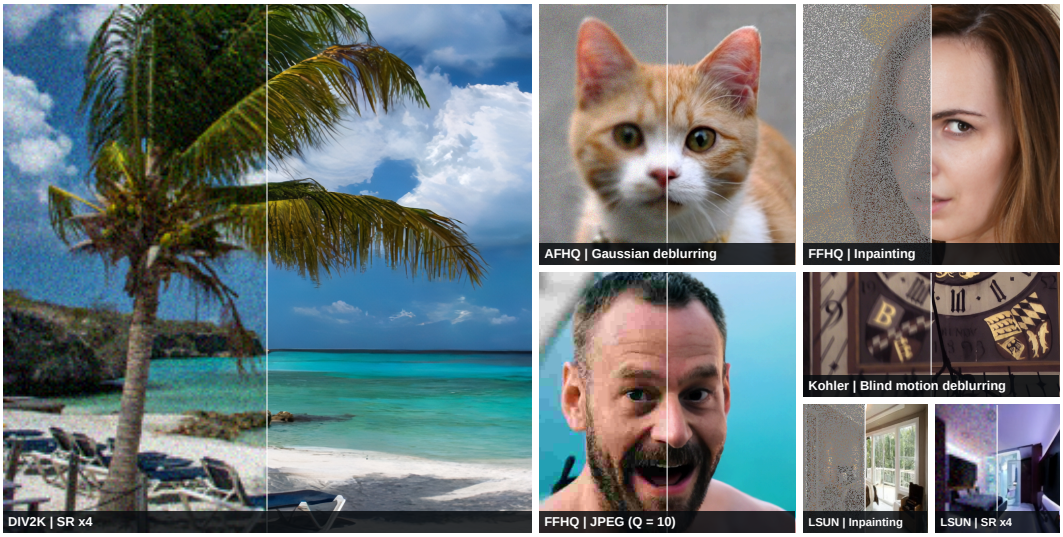

Figure 1: **BAM! One small network, many imaging problems, few steps.** Each tile pairs an observation (left) with a BAM! posterior sample (right). A single 36M-parameter network, with the forward operator supplied at inference time, covers $\times 4$ super-resolution (DIV2K, LSUN), Gaussian deblurring (AFHQ), inpainting (FFHQ, LSUN), non-linear JPEG restoration at $Q = 10$ (FFHQ) and blind motion deblurring (Köhler), all in 3 steps. Settings are given in Section 4.

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.

1 INTRODUCTION

We consider imaging problems involving an unknown image $x^\star \in \mathbb{R}^d$ and a measurement $y \in \mathbb{R}^m$, related through the observation model $y = Ax^\star + \sigma_y w$, where w is additive Gaussian noise and the forward operator $A \in \mathbb{R}^{m\times d}$ and the noise level $\sigma_y > 0$ are known at inference time. We are interested in problems where estimating $x^\star$ from y is ill-posed or ill-conditioned. Additional information is then required to reduce the uncertainty about $x^\star$ and deliver meaningful solutions. Within the Bayesian framework, this is achieved by modelling $x^\star$ as a realization of $\boldsymbol{x} \sim p(\boldsymbol{x})$, the so-called prior distribution, and y as a realisation from $p(\boldsymbol{y} \mid x^\star)$. Prior and observed information are then combined in the posterior distribution $p(\boldsymbol{x} \mid y, A, \sigma_y) = p(y \mid \boldsymbol{x}, A, \sigma_y)\, p(\boldsymbol{x})/p(y \mid A, \sigma_y)$. This posterior underpins Bayesian inference about $\boldsymbol{x}$, from point estimators such as the posterior mean to uncertainty quantification through posterior variances and credible regions, all of which can be approximated from posterior samples.

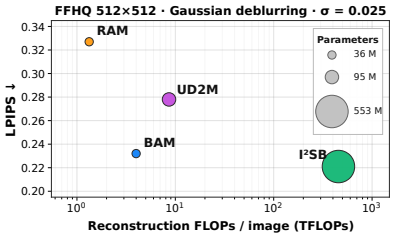

Figure 2: Pareto frontier drawn by SOTA methods compared to BAM.

State-of-the-art Bayesian imaging methods learn from representative data via plug-and-play (PnP) or end-to-end strategies. PnP methods plug a pre-trained denoiser or generative prior into an iterative scheme that enforces the likelihood (Venkatakrishnan et al., 2013; Romano et al., 2017; Kamilov et al., 2023), and zero-shot posterior samplers extend this idea beyond point estimates (Laumont et al., 2022; Kawar et al., 2022; Chung et al., 2023). With modern diffusion model (DM) priors (Ho et al., 2020; Song et al., 2021), this flexibility carries three costs: a biased likelihood approximation with tunable guidance weights, hundreds of neural function evaluations (NFEs), and priors that, even in latent space, reach billions of parameters (Podell et al., 2024). Distilled consistency and flow-map models (Song et al., 2023; Luo et al., 2023; Boffi et al., 2025) cut sampling to a few NFEs, and zero-shot solvers built on distilled latent models now set the state of the art (Garber & Tirer, 2025; Spagnoletti et al., 2025; 2026), but the biased likelihood and large prior remain. End-to-end methods instead learn the reconstruction directly, typically with physics-aware unrolled or equilibrium networks (Monga et al., 2021; Gilton et al., 2021) trained per operator, often by fine-tuning a pre-trained denoiser, returning point estimates. Conditional diffusion and bridge models sample the posterior (Liu et al., 2023a; Zhao et al., 2024; Mbakam et al., 2025) but are mostly task-specific and as costly as the underlying DM. Since y is highly informative about $\boldsymbol{x}$, posterior sampling should need far fewer parameters than unconditional generation, yet lightweight networks that generalize across operators remain largely unexplored. A notable exception is the Reconstruct Anything Model (RAM) (Terris et al., 2026), an operator-conditioned DRUNet that solves many linear inverse problems with one network, but returns a point estimate of $\boldsymbol{x}$ not posterior samples.

We introduce the Bayesian Anything Model (BAM), a lightweight foundation model for few-step, physics-aware posterior sampling that bridges these two camps. BAM is a conditional flow map that transports a Gaussian reference to the posterior $p(\boldsymbol{x} \mid y, A, \sigma_y)$, with the instrument physics specified at inference time rather than fixed during training. Pre-trained jointly on large image corpora and libraries of forward operators, a single model draws posterior samples in a few steps, with no likelihood approximation and no guidance weights to tune. Our contributions are as follows:

- **A flow-map training paradigm.** We propose a recipe that upgrades an unfolded reconstruction network into such a flow map, via Lagrangian self-distillation on a normalized stochastic interpolant of the measurement.
- **A lightweight foundation model.** Built on the 36M-parameter RAM backbone, a UNet with unrolled physics-aware updates, BAM learns this map across operators, noise levels and datasets, delivering high-quality posterior samples (Figure 1) and generalizing to new data and tasks.
- **State-of-the-art quality at a fraction of the cost.** On FFHQ, AFHQ, LSUN, DIV2K and the Köhler benchmark, BAM outperforms specialized models and leading zero-shot methods in sample quality in just three steps, advancing the quality–cost Pareto frontier (Figure 2).

BAM lowers the barrier to generative computational imaging, cuts the economic and environmental cost of training imaging models, and opens a new direction for physics-aware Bayesian imaging.

2 BACKGROUND

Diffusion models. DMs generate samples from ρ_0 by learning to reverse a process that gradually corrupts data with Gaussian noise. The noising process is $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{z}$, where $\boldsymbol{x}_0 \sim \rho_0$ and $\boldsymbol{z} \sim \mathcal{N}(0, \sigma_d^2 \mathrm{Id})$ are independent, and the schedules satisfy $\alpha_0 = \sigma_1 = 1$ and $\alpha_1 = \sigma_0 = 0$. Each marginal ρ_t, the law of $\boldsymbol{x}_t$, is therefore a rescaled, Gaussian-smoothed version of the data distribution. Sampling reverses this process through the probability-flow ODE (Song et al., 2021)

$$\dot{x}_t = f_t x_t - \tfrac{1}{2} g_t^2 \nabla \log \rho_t(x_t), \qquad f_t = \frac{\dot{\alpha}_t}{\alpha_t}, \quad g_t^2 = 2\sigma_d^2 \, \alpha_t \sigma_t \frac{\mathrm{d}}{\mathrm{d}t}\Big(\frac{\sigma_t}{\alpha_t}\Big). \tag{1}$$

Integrated backwards in time, its solutions transport ρ_1 onto ρ_t for every $t \in [0,1]$. The associated integrated flow $X_{t,0}$ sends each state to its endpoint at $t = 0$, so that $(X_{t,0})_{\#}\rho_t = \rho_0$. DMs learn only the score $\nabla \log \rho_t$, which makes them easy to train but slow to sample. A standard DM solver needs hundreds of NFEs to integrate (1) for each sample.

Flow matching. Flow matching (FM) learns the velocity field of the probability-flow ODE directly, rather than through the score (Lipman et al., 2023; Liu et al., 2023b; Albergo et al., 2025). The drift that transports the marginals $(\rho_t)_{t\in[0,1]}$ is the average velocity of the noising paths through each state,

$$b_t(x) = \mathbb{E}\big[\dot{\alpha}_t \boldsymbol{x}_0 + \dot{\sigma}_t \boldsymbol{z} \,\big|\, \boldsymbol{x}_t = x\big], \tag{2}$$

so the probability flow reads $\dot{x}_t = b_t(x_t)$. The drift b_t minimises $\mathbb{E}\|v(\boldsymbol{x}_t) - \dot{\alpha}_t \boldsymbol{x}_0 - \dot{\sigma}_t \boldsymbol{z}\|^2$ over functions v, so it can be learned by least-squares regression on $(\boldsymbol{x}_0, \boldsymbol{z})$ pairs without simulating ODE trajectories. FM typically uses $\alpha_t = 1 - t$ and $\sigma_t = t$, for which $b_t(x) = \mathbb{E}[\boldsymbol{z} - \boldsymbol{x}_0 \mid \boldsymbol{x}_t = x]$. By Tweedie's formula, $b_t(x) = f_t x - \frac{1}{2} g_t^2 \nabla \log \rho_t(x)$, the r.h.s. of (1). So FM and DMs learn the same ODE and differ only in parametrization, loss weighting and choice of solver (Gao et al., 2025).

Consistency models. Consistency models (CMs) dramatically accelerate sampling by learning the integrated flow $X_{t,0}$ directly, mapping a noisy state to the endpoint of its trajectory in a single network evaluation. Training enforces two properties of this map. The first is *self-consistency*: all points on the same trajectory of (1) share an endpoint, so $X_{t,0}(x_t) = X_{s,0}(x_s)$. The second is the boundary condition $X_{0,0}(x) = x$. CMs can be distilled from a pre-trained DM or trained directly from data (Song et al., 2023; Song & Dhariwal, 2024; Boffi et al., 2025).

Flow maps. Flow maps generalize CMs to jumps between any two times (Boffi et al., 2025). The flow map $X_{t,s}$ moves a point along its trajectory of (1) from time t to time $s \le t$, i.e. $X_{t,s}(x_t) = x_s$. Thus $X_{t,t}(x) = x$, $X_{t,0}$ is the CM map, and $X_{1,0}$ transports pure noise to data. The parametrization

$$X_{t,s}(x) = x + (s - t)\, v_{t,s}(x) \tag{3}$$

enforces $X_{t,t}(x) = x$ by construction, and $v_{t,s}$ is the average velocity along the trajectory between t and s. On the so-called diagonal ($t = s$), it reduces to the FM drift:

$$\partial_s X_{t,s}(x)\big|_{s=t} = v_{t,t}(x) = b_t(x). \tag{4}$$

Under mild regularity, flow maps verify the following characterizations (Boffi et al., 2025, Prop. 2.2):

$$\underbrace{\partial_s X_{t,s}(x) = b_s\big(X_{t,s}(x)\big)}_{\text{Lagrangian}}, \qquad \underbrace{\partial_t X_{t,s}(x) + DX_{t,s}(x)\, b_t(x) = 0}_{\text{Eulerian}}, \qquad \underbrace{X_{u,s} \circ X_{t,u} = X_{t,s}}_{\text{semigroup}}. \tag{5}$$

for $s \le u \le t$, where $DX_{t,s}$ denotes the Jacobian. The three characterizations are equivalent, as each determines the flow map uniquely. With $s = 0$, the semigroup property is exactly CM self-consistency, and the Eulerian equation is its infinitesimal form. The Lagrangian form avoids the Jacobian, which makes it cheaper and more stable to train.

For a network $X^\theta_{t,s}(x) = x + (s - t)\, v^\theta_{t,s}(x)$, the tangent and Lagrangian conditions give two losses:

$$\mathcal{L}_b(\theta) = \mathbb{E}\big\|v^\theta_{\boldsymbol{t},\boldsymbol{t}}(\boldsymbol{x_t}) - \dot{\alpha}_{\boldsymbol{t}} \boldsymbol{x}_0 - \dot{\sigma}_{\boldsymbol{t}} \boldsymbol{z}\big\|^2, \qquad \mathcal{L}_{\mathrm{LSD}}(\theta) = \mathbb{E}\big\|\partial_s X^\theta_{\boldsymbol{t},\boldsymbol{s}}(\boldsymbol{x_t}) - v^{\theta^-}_{\boldsymbol{s},\boldsymbol{s}}\big(X^\theta_{\boldsymbol{t},\boldsymbol{s}}(\boldsymbol{x_t})\big)\big\|^2. \tag{6}$$

The times $\boldsymbol{s} \le \boldsymbol{t}$ are drawn at random, and θ^- is an exponential moving average (EMA) of θ that serves as a detached teacher. $\mathcal{L}_b$ is the FM loss on the diagonal, whereas the Lagrangian self-distillation loss $\mathcal{L}_{\mathrm{LSD}}$ propagates that drift to finite jumps.

We use this formulation to train BAM, our lightweight foundation model for few-step, physics-aware posterior sampling, which transfers to unseen data and tasks zero-shot or with minimal finetuning.

3 BAM! BAYESIAN ANYTHING MODEL

Model. BAM is designed for imaging problems of the form $y = Ax^\star + \sigma_y w$, where we assume that the unknown image $x^\star$ is a realization of a r.v. $\boldsymbol{x} \sim p(\boldsymbol{x})$, the additive noise w is a realization of $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathrm{Id})$, and the forward operator $A \in \mathbb{R}^{m\times d}$ and noise level $\sigma_y > 0$ are known at inference time. The goal is to draw accurate samples from the posterior $p(\boldsymbol{x} \mid y, A, \sigma_y)$ in a small number of network evaluations. Moreover, rather than train a separate model for each problem or narrow problem class, BAM amortizes over a wide family of operators and noise levels encountered in practice, which are randomized during training. A single model therefore serves the entire family, requiring no retraining, or at most light finetuning, when the forward model changes. Finally, following Terris et al. (2026), BAM is deliberately lightweight, compatible with modest local hardware.

To this end, BAM is built as a lightweight few-step conditional flow map targeting the posterior $p(\boldsymbol{x} \mid y, A, \sigma_y)$. Feeding $(y, \mathcal{A}, \sigma_y, t, s)$ directly as inputs to a small network is numerically fragile, as σ_y and (t, s) span several orders of magnitude while the spectrum of $\mathcal{A}$ is fixed, so the network receives inputs whose relative scales vary inconsistently across the training range. We therefore condition on a rescaled measurement, defined through the stochastic interpolant

$$\boldsymbol{y}_\sigma = \alpha_\sigma A\boldsymbol{x} + \varsigma_\sigma \boldsymbol{w}, \tag{7}$$

with schedule α_σ and ς_σ, respectively decreasing and increasing in σ, satisfying $\varsigma_\sigma/\alpha_\sigma = \sigma$. Hence y is a realization of $\boldsymbol{y}_\sigma$ at $\sigma = \sigma_y$, up to the known factor α_σ, and we condition on $y_\sigma := \alpha_\sigma y$.

Using the linear interpolant of section 2, we construct a flow map to transport $\rho_1 = \mathcal{N}(\mathbf{0}, \sigma_d^2 \mathrm{Id})$ to $\rho_0 = p(\boldsymbol{x} \mid y, A, \sigma_y)$. We parametrize this flow map as

$$\boldsymbol{X}^\theta_{t,s}(x \mid y_\sigma, \alpha_\sigma A) = x + (s-t)\, \boldsymbol{v}_\theta(x, t, s, y_\sigma, \alpha_\sigma A), \qquad 0 \le s \le t \le 1, \tag{8}$$

so that the identity boundary condition $\boldsymbol{X}^\theta_{t,t}(x \mid y_\sigma, \alpha_\sigma A) = x$ holds $\forall x \in \mathbb{R}^d$ by construction. The velocity must then satisfy two conditions. On the diagonal, the tangent condition requires $\boldsymbol{v}_\theta(x, t, t, y_\sigma, \alpha_\sigma A) = \boldsymbol{b}_t(x \mid y_\sigma, \alpha_\sigma A)$, the conditional probability-flow drift. Off the diagonal, backwards jumps from t to s are governed by the Lagrangian condition of section 2, applied conditionally for posterior sampling. Together these two conditions characterize the conditional flow map and lead to BAM's two main training objectives. Note that σ_y is not supplied to the network in (8). It enters only through the rescaling (7), so $\boldsymbol{v}_\theta$ acts partially blindly w.r.t. σ_y. Since σ_y is easily estimated from y, dropping the explicit conditioning costs little and removes an embedding pathway, keeping the architecture lightweight.

The network $\boldsymbol{v}_\theta$ can be realized by any physics-aware architecture for linear imaging problems, notably deep unfolding architectures, by conditioning on the augmented measurement $[x_t\,;\, y_\sigma]$ and the stacked forward operator $[(1-t)\mathrm{Id}\,;\, \alpha_\sigma A]$ supplied at inference time. For BAM we adopt a RAM backbone, which embeds the unfolding philosophy in a DRUNet, and assign its two noise-level embeddings to input t and s (see sections B and D.2 for architecture details and ablations).

Training. As mentioned previously, BAM relies on the following two flow map training objectives. On the diagonal ($t = s$), the velocity is fitted to the interpolant slope by minimizing

$$\mathcal{L}_b(\theta) = \mathbb{E}\Big[\big\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, \boldsymbol{t}, \boldsymbol{t}, \boldsymbol{y}_\sigma, \boldsymbol{A}) - (\boldsymbol{z} - \boldsymbol{x}_0)\big\|_2^2\Big]. \tag{9}$$

Off diagonal ($s < t$), holding the flow's starting point (x_t, t) and the conditioning $(y_\sigma, \sigma A)$ fixed, the endpoint derivative of (8) is

$$\partial_s \boldsymbol{X}^\theta_{t,s} = \boldsymbol{v}_\theta(x_t, t, s, y_\sigma, \alpha_\sigma A) + (s-t)\, \partial_s \boldsymbol{v}_\theta(x_t, t, s, y_\sigma, \alpha_\sigma A), \tag{10}$$

which we compare with the diagonal velocity of the EMA teacher at the transported point, i.e.,

$$\mathcal{L}_{\mathrm{LSD}}(\theta) = \mathbb{E}\Big[\big\|\partial_s \boldsymbol{X}^\theta_{\boldsymbol{t},\boldsymbol{s}} - \mathrm{sg}\big[\boldsymbol{v}_{\theta^-}(\hat{\boldsymbol{x}}_{\boldsymbol{t},\boldsymbol{s}}, \boldsymbol{s}, \boldsymbol{s}, \boldsymbol{y}_\sigma, \boldsymbol{A})\big]\big\|_2^2\Big], \quad \hat{\boldsymbol{x}}_{\boldsymbol{t},\boldsymbol{s}} := \boldsymbol{X}^\theta_{\boldsymbol{t},\boldsymbol{s}}(\boldsymbol{x}_t \mid \boldsymbol{y}_\sigma, \boldsymbol{\sigma A}). \tag{11}$$

BAM is trained on the flow objective $\mathcal{L}_b + \lambda \mathcal{L}_{\mathrm{LSD}}$, with the expectations taken over $\boldsymbol{x_t} = (1-t)\boldsymbol{x} + t\boldsymbol{z}$ with $\boldsymbol{x} \sim p_{\boldsymbol{x}}$ and $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathrm{Id})$ independent of $\boldsymbol{x}$, $\boldsymbol{y} \sim N(\boldsymbol{Ax}, \sigma^2 \mathrm{Id})$ with $\boldsymbol{A} \sim p_{\boldsymbol{A}}$ independent of $(\boldsymbol{x}, \boldsymbol{z})$, and $\boldsymbol{t} \sim \mathcal{U}[0, 1]$ and $\boldsymbol{s}|\boldsymbol{t}$ from (17). Rather than draw σ independently, we tie it to $\boldsymbol{t}$ during training through $\sigma(t) = \sigma_{\max}\gamma t/(1-(1-\gamma)t)$, which increases from 0 to $\sigma_{\max}$ with shape controlled

by $\gamma > 0$. This concentrates the capacity of the network on the regime where $\boldsymbol{y}_\sigma$ and $\boldsymbol{x}_t$ carry broadly comparable amounts of noise, a challenging regime where the flow must bend to take both y_σ and x_t into account. Note that sg in (11) stops gradients through the complete teacher evaluation, including its input, and $\partial_s \boldsymbol{X}^\theta_{t,s}$ is evaluated by an automatic-differentiation Jacobian–vector product that retains the graph needed to optimize it.

The above flow objective is paired with two auxiliary terms, targeting perceptual quality and contrast bias, which act near $s \approx 0$. More precisely, the full objective for BAM is

$$\mathcal{L}(\theta) = \mathbb{E}\Big[\lambda_b \ell_b + \lambda_L \ell_{\mathrm{LSD}} + g(s)\big\{\lambda_p \ell_{\mathrm{LPIPS}}(\hat{\boldsymbol{x}}_{t,s}, \boldsymbol{x}_0) + \lambda_c \ell_{\mathrm{ctr}}(\hat{\boldsymbol{x}}_{t,s}, \boldsymbol{x}_0)\big\}\Big], \tag{12}$$

where $g(s) = \exp(-4s)$ assigns weight to the auxiliary losses only when s is small, ℓ_{LPIPS} is the squeeze-based perceptual loss of Zhang et al. (2018), applied to the clipped output, and ℓ_{ctr} compares intensity histograms to penalize contrast bias. The latter provides robustness when training across datasets with differing image statistics, and we set $\lambda_c = 0$ after pre-training. No adversarial loss is used at any stage. Sections C, D.3 and D.4, detail the loss functions, their optimization and ablations.

Posterior sampling. Since BAM is trained on the locus where the noise in $\boldsymbol{y}_\sigma$ and in $\boldsymbol{x}_t$ is balanced, each sampling step should pair the current iterate $\boldsymbol{x}_t$ with a draw of the interpolant $\boldsymbol{y}_\sigma$ at the matching level σ. For $\sigma > \sigma_y$, such a draw consistent with y is obtained by adding the missing noise,

$$\boldsymbol{y}_\sigma = \alpha_\sigma y + \sqrt{\varsigma_\sigma^2 - \alpha_\sigma^2 \sigma_y^2}\, \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathrm{Id}), \tag{13}$$

which has noise variance ς_σ^2. We hold $\boldsymbol{\varepsilon}$ fixed across steps. Sampling follows a decreasing schedule $1 = t_1 > t_2 > t_3$, each step noising the current iterate to level t_k and mapping it back to $s = 0$,

$$\boldsymbol{x}_{t_k} = (1 - t_k)\hat{\boldsymbol{x}}^{(k-1)} + t_k \boldsymbol{z}^{(k)}, \qquad \boldsymbol{z}^{(k)} \sim \mathcal{N}(\mathbf{0}, \sigma_d^2 \mathrm{Id}), \tag{14}$$

$$\hat{\boldsymbol{x}}^{(k)} = \boldsymbol{x}_{t_k} - t_k\, \boldsymbol{v}_\theta(\boldsymbol{x}_{t_k}, t_k, 0, \boldsymbol{y}_{\sigma_k}, \sigma_k A), \tag{15}$$

where $\sigma_k = \sigma(t_k)$ and $t_1 = 1$, so that $\boldsymbol{x}_{t_1} = \boldsymbol{z}^{(1)}$ is pure noise and (14) requires no previous iterate.

When σ_y is small, the intermediate level satisfies $\sigma_2 \in (\sigma_y, \sigma_1)$ and the final step is taken at $\sigma_3 = \sigma_y$, where no re-noising of the measurement is required. When σ_y is large, ending at σ_y would leave a long final jump and thus introduce some sampling bias. Hence, we instead set $\sigma_2 = \sigma_y$ and take the last step at $\sigma_3 < \sigma_y$, drawing the interpolant by a Brownian bridge between $\boldsymbol{A}\hat{\boldsymbol{x}}^{(2)}$ and the rescaled measurement. Each prediction costs one network evaluation and applies $\boldsymbol{X}^\theta_{t_k,0}$ rather than composing consecutive maps. Randomness enters through the initialization, (13), and the re-noising, so independent runs give independent posterior samples.

4 Experiments

4.1 Setup

Datasets. BAM is trained on 4KLSDB (Zhu et al., 2026), bicubically downsampled to 2K resolution, together with HQ-50K (Yang et al., 2023), DIV2K (Agustsson & Timofte, 2017), FFHQ-512 (Karras et al., 2019), and the LSDIR (Li et al., 2023) images above 1K resolution. Pixel values are scaled to $[0, 1]$.[1] We evaluate on held-out test sets of 300 LSUN Bedroom (Yu et al., 2015) images, 64 FFHQ and 64 AFHQ cat (Choi et al., 2020) images, and between 64 and 100 DIV2K images depending on the method, at noise levels $\sigma_y \in \{0.025, 0.05\}$.

Metrics. We report PSNR (↑) for pixel-level fidelity, vgg LPIPS (Zhang et al., 2018) (↓) for perceptual similarity, and CMMD (Jayasumana et al., 2024) (↓) and FID (Heusel et al., 2017) (↓) for distributional similarity to the ground truth. FID is unstable for small test sets and is reported for comparability with prior work; CMMD, designed to address this, should be considered the more reliable of the two.

[1] Internally, the network rescales images to $[-1, 1]$ and the noise level to $2\sigma_y$.

Training. Starting from the public RAM checkpoint[2], we train the baseline BAM for 170 hours on 8 H100 GPUs, jointly on all inverse problems. These are bicubic SR$\times$4 and $\times$8, isotropic Gaussian blur with $\sigma_{\text{blur}} \in [1, 5]$, motion blur with random 61×61 kernels of intensity 0.5[3], compressed sensing at a 25% rate, inpainting with 80% of pixels masked at random, and demosaicing, all with noise levels $\sigma_y \in [0, 0.05]$. The fine-tuned variant, BAM*, continues from this baseline for 16 hours on 2 H100 GPUs on each target dataset, still across all problems (see section C).

Baselines. We compare BAM and BAM* with zero-shot methods (LATINO, LATINO-PRO, TReg) and training-based methods (RAM, SILO, UD2M, I^{2}SB), described in Appendix A. UD2M and I^{2}SB are pixel-space diffusion models limited to lower resolutions, so we evaluate them only on FFHQ-512. SILO, UD2M and I^{2}SB use one checkpoint per dataset–problem pair (although UD2M can be trained over operator families), and RAM is evaluated before and after finetuning. All are fine-tuned with their public code, at the evaluated noise levels, for the cumulative runtime of one BAM dataset finetuning, whereas a single BAM model covers all noise levels.

4.2 Results

Reconstruction quality. Table 1 compares all methods on DIV2K and FFHQ at $\sigma_y = 0.05$. finetuning improves the LPIPS of three-step BAM on FFHQ for all five problems at both noise levels, but slightly degrades it on DIV2K (tables 5 and 6), indicating that the extended training set helps the network generalize to diverse, high-resolution data, while more focused finetuning on a specific distribution such as FFHQ improves results there. Tables 4–7 in Appendix E report all problems on which BAM is trained, and include the out-of-distribution datasets AFHQ and LSUN bedroom, where finetuning noticeably improves results and the baseline BAM remains competitive with the other methods. Perceptually, BAM stands out: across the four datasets, five problems and two noise levels reported in Appendix E, a BAM variant attains the best LPIPS in 38 of 40 settings and the best CMMD in 36 of 40, the exceptions being FFHQ deblurring (I^{2}SB) and one super-resolution setting each on AFHQ (SILO) and LSUN (UD2M). Representative restorations are shown in figs. 4 to 6: BAM restores sharp and realistic textures such as fur, hair and foliage while remaining consistent with the measurements, whereas RAM, trained as an MMSE estimator, often attains higher PSNR at the cost of smoother textures, and zero-shot and latent-space methods exhibit noise artifacts or hallucinate content that is inconsistent with the observation.

Computation/quality trade-off. Figure 2 reports image quality (LPIPS), computational cost (floating-point operations to restore a 512×512 input) and model size (number of parameters) for BAM* and the state-of-the-art baselines in a single graph. We observe that BAM* advances the Pareto frontier of the baselines, attaining better quality at a lower cost, and the same conclusion holds across metrics, inverse problems and datasets (Table 1 and Appendix E). BAM is also robust to post-training quantization: converting it to INT8 under TensorRT, together with a TensorRT-friendly reformulation of the physics-aware operations, cuts the latency on a 512×512 image from 281 ms to 80 ms on an NVIDIA A40 GPU ($3.5\times$ faster), with only a moderate loss in accuracy (PSNR -1.3 dB, LPIPS $0.400 \to 0.427$) and *without any fine-tuning after quantization* (Appendix F). Quantization-aware fine-tuning is expected to narrow the remaining gap.

Other experiments. Appendix E.3 extends BAM to sparse-view CT on LIDC-IDRI chest slices with 51 parallel-beam projections, following the setting of Terris et al. (2026), a single-channel modality far from the natural images seen during pre-training. Figure 3 shows the ground truth, the adjoint reconstruction $\mathcal{A}^\dagger \boldsymbol{y}$, a BAM sample, and the 4×4-block standard-deviation and residual maps. After finetuning on CT, three-step BAM samples recover fine lung vessels that the RAM estimate smooths out. Moreover, the pixelwise standard deviation over 64 draws concentrates on anatomical edges and vessels, and is of the same order of magnitude as the error of the empirical posterior mean, so BAM provides a spatial uncertainty map at no extra training cost.

Appendix E.4 turns to blind motion deblurring with spatially varying kernels. Following Terris et al. (2026), we estimate $\boldsymbol{A}$ with the pre-trained Kernel Predictor Network (KPN) of Carbajal et al. (2023)

[2] https://github.com/matthieutrs/ram

[3] https://github.com/LeviBorodenko/motionblur; see Appendix E.4.

Table 1: **DIV2K and FFHQ restoration at $\sigma_y = 0.05$.** PSNR (dB; ↑), LPIPS (↓), CMMD (↓), and FID (↓). BAM variants use thre e steps; BAM* and RAM* denote finetuning. Bold (underline) mark the best (second-best) reported values for each dataset and problem; – denotes an unavailable result.

Method	NFEs	DIV2K				FFHQ			
		PSNR↑	LPIPS↓	CMMD↓	FID↓	PSNR↑	LPIPS↓	CMMD↓	FID↓
Gaussian deblurring									
BAM	3	24.48	**0.35**	0.06	**34.4**	29.63	0.25	0.10	45.0
BAM*	3	24.44	**0.35**	**0.04**	35.6	30.16	0.25	0.08	51.6
RAM	1	25.63	0.43	0.44	52.6	30.54	0.35	1.15	94.7
RAM*	1	**26.00**	0.42	0.34	47.0	**31.53**	0.34	1.07	92.4
LATINO-PRO	65	23.55	0.54	0.49	71.9	23.40	0.46	1.17	128.3
SILO	200	17.20	0.62	1.84	298.8	24.92	0.38	0.25	91.8
I^{2}SB	50	–	–	–	–	30.24	**0.24**	**0.02**	**44.2**
UD2M	12	–	–	–	–	29.30	0.29	0.72	67.3
Super-resolution									
BAM	3	24.30	**0.36**	**0.08**	**39.6**	28.84	**0.27**	**0.19**	**50.4**
BAM*	3	23.56	0.37	0.09	41.1	29.37	**0.27**	**0.19**	62.4
RAM	1	25.79	0.43	0.26	74.1	**31.15**	0.36	1.02	107.1
RAM*	1	**25.88**	0.42	0.27	51.7	31.01	0.35	1.52	93.0
LATINO-PRO	65	22.82	0.54	0.59	95.3	23.18	0.53	0.90	146.3
SILO	200	16.53	0.64	2.04	308.4	25.04	0.37	0.20	80.7
I^{2}SB	50	–	–	–	–	26.82	0.39	1.02	148.8
UD2M	12	–	–	–	–	28.16	0.31	0.30	82.2
Inpainting									
BAM	3	26.61	**0.27**	**0.02**	**20.5**	31.50	0.23	0.08	**32.1**
BAM*	3	25.66	**0.27**	0.03	21.0	31.57	**0.20**	**0.05**	32.4
RAM	1	23.52	0.41	0.30	66.9	25.55	0.39	1.53	126.5
RAM*	1	**27.61**	0.32	0.18	35.5	**32.40**	0.29	1.12	56.5
LATINO-PRO	65	11.25	0.71	2.01	282.4	13.44	0.79	2.63	305.0
SILO	200	12.92	0.73	2.91	382.4	17.46	0.52	1.00	139.4
I^{2}SB	50	–	–	–	–	18.89	0.58	1.93	174.2
UD2M	12	–	–	–	–	29.29	0.25	0.21	45.54

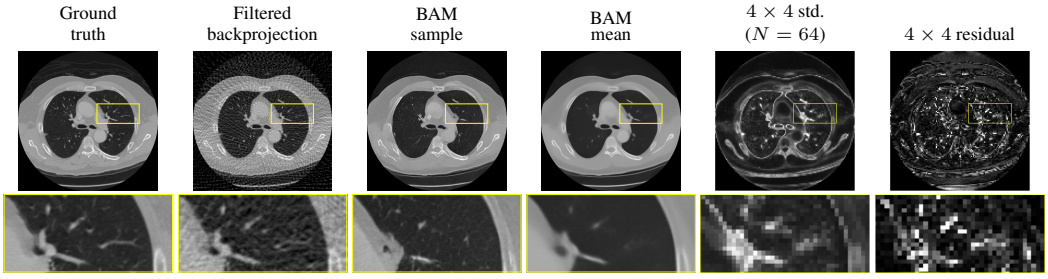

Figure 3: CT reconstruction and 4×4-block uncertainty and residual maps. Left to right: ground truth, reconstruction $\mathcal{A}^\dagger \boldsymbol{y}$, one BAM sample, posterior mean, standard deviation over 64 draws, and residual $|\mathrm{GT} - \mathrm{mean}|$. Yellow boxes and $4\times$ strips show the same lung region in every panel.

and solve the resulting inverse problem. Although BAM has never been trained on such operators, it outperforms both RAM and the original unrolled PnP architecture of Carbajal et al. (2023) on the Köhler dataset (Köhler et al., 2012) (see Figure 21). Fine-tuning BAM jointly with the KPN, as in the state of the art, is a natural next step towards a generative counterpart of their method.

Appendix E.5 reports a preliminary exploration of non-linear imaging problems, namely the restoration of noisy, JPEG-compressed images. Zero-shot and unrolled methods generalize poorly here, as they require a quadratic approximation of the log-likelihood. After finetuning on this problem, BAM instead generates the missing details and regularizes noisy areas. This approach has limitations, and extending BAM to more general degradation models is a natural direction for future work.

Appendix E.6 reports a preliminary assessment of BAM from a distortion-perception viewpoint, based on repeated draws for a fixed problem. Drawing multiple realizations from a single observation $\boldsymbol{y}$ and averaging them gives a Monte Carlo estimate of the posterior mean, which we compare with the MMSE estimator produced by RAM. The empirical mean exceeds the PSNR of RAM* on four of the five problems, the exception being super-resolution. The draws are therefore diverse enough to be perceptually sharp individually, while their average remains an accurate posterior mean estimate.

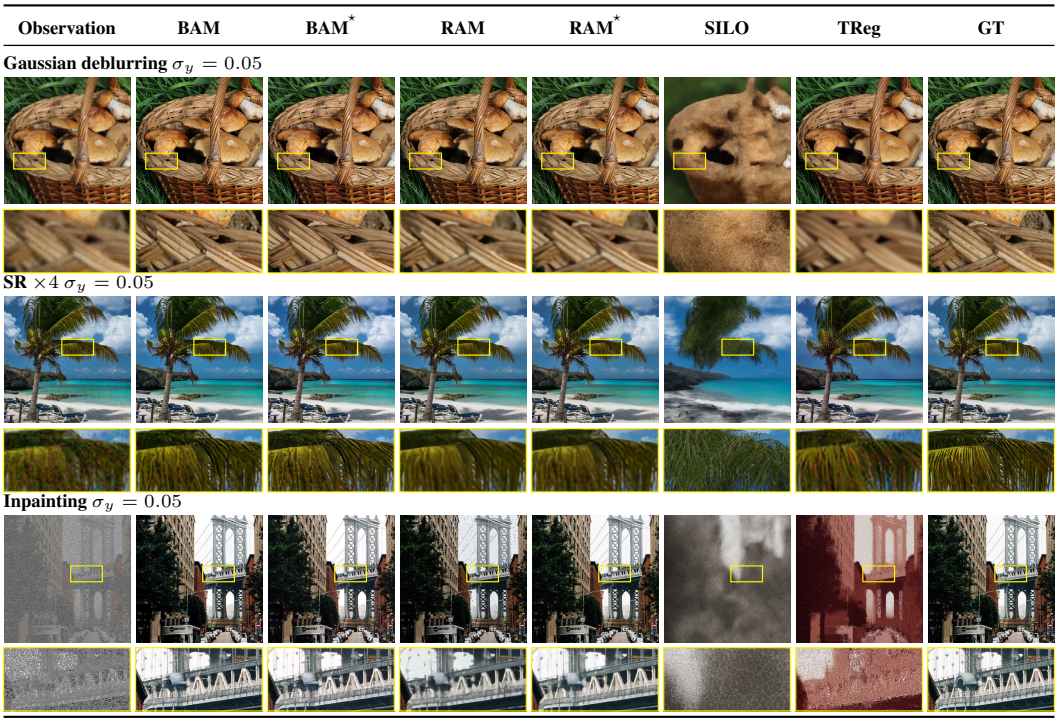

Figure 4: **DIV2K restorations** at $\sigma_y = 0.05$, one example per problem. BAM and BAM* use three steps. Strips show $4\times$ zoom. More problems and comparisons in Figure 16.

5 Conclusion and Limitations

We introduced BAM, a 36M-parameter physics-aware generative foundation model for Bayesian computational imaging that generalizes robustly to unseen data and tasks, zero-shot or with minimal finetuning. BAM upgrades the operator-conditioned RAM backbone into a conditional flow map or consistency model, pre-trained jointly on large image corpora and libraries of forward operators. The forward operator is an input to the network, so instrument physics is specified at inference time. A single lightweight network then draws posterior samples without likelihood approximations, guidance weights or task-specific retraining. Across linear inverse problems on FFHQ, AFHQ, LSUN, DIV2K and Köhler, BAM outperformed both specialised models and leading zero-shot methods in just 3 steps, at a fraction of their computational cost.

BAM has four main limitations. It handles only additive Gaussian noise and linear or mildly nonlinear forward operators. It is non-blind, i.e., the instrument model must be supplied, so blind problems rely on an external estimate of the operator. It does not detect model misspecification, so under strong distribution shift it can return unreliable posteriors without warning. Finally, posterior sample quality was assessed empirically, and formal guarantees for the learned conditional flow map remain open.

BAM brings generative Bayesian imaging within reach of the global computational imaging community. The natural next applications are scientific and medical imaging, such as MRI, CT, microscopy and astronomy, where forward operators are known and ground truth is scarce. In these domains, posterior samples support Bayesian decision theory, optimal estimation, uncertainty quantification and experimental design, adding significant value to existing point estimation approaches. On the

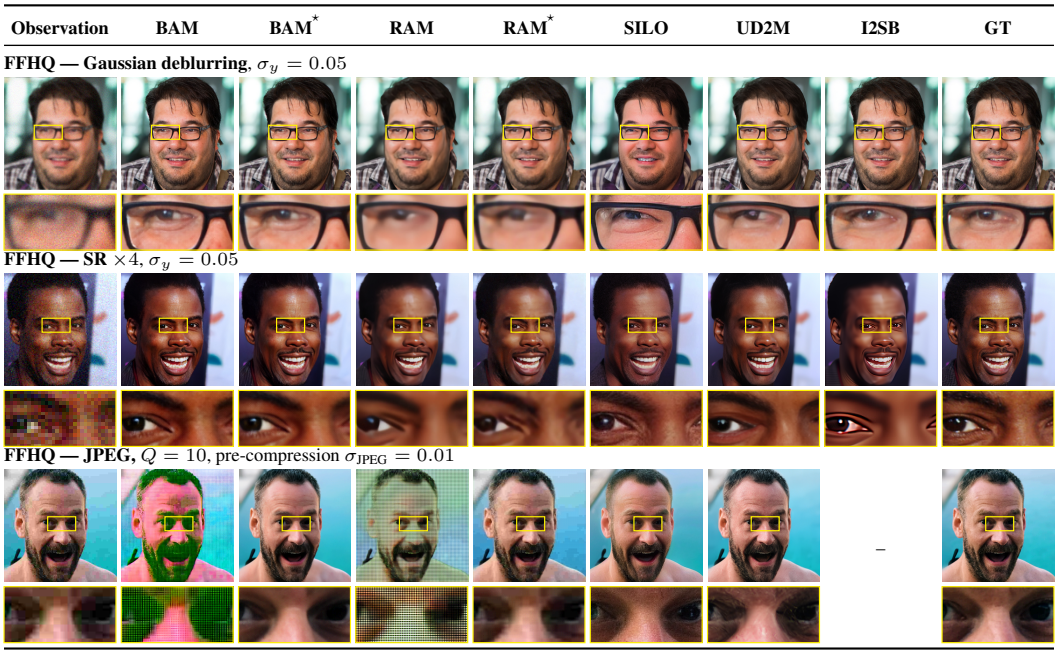

Figure 5: **FFHQ restorations.** Deblurring and super-resolution at $\sigma_y = 0.05$; JPEG at $Q = 10$ with pre-compression noise $\sigma_{\text{JPEG}} = 0.01$. BAM and BAM* use three steps. Strips show $4\times$ zoom; – marks unavailable results. More problems and comparisons in Figure 17.

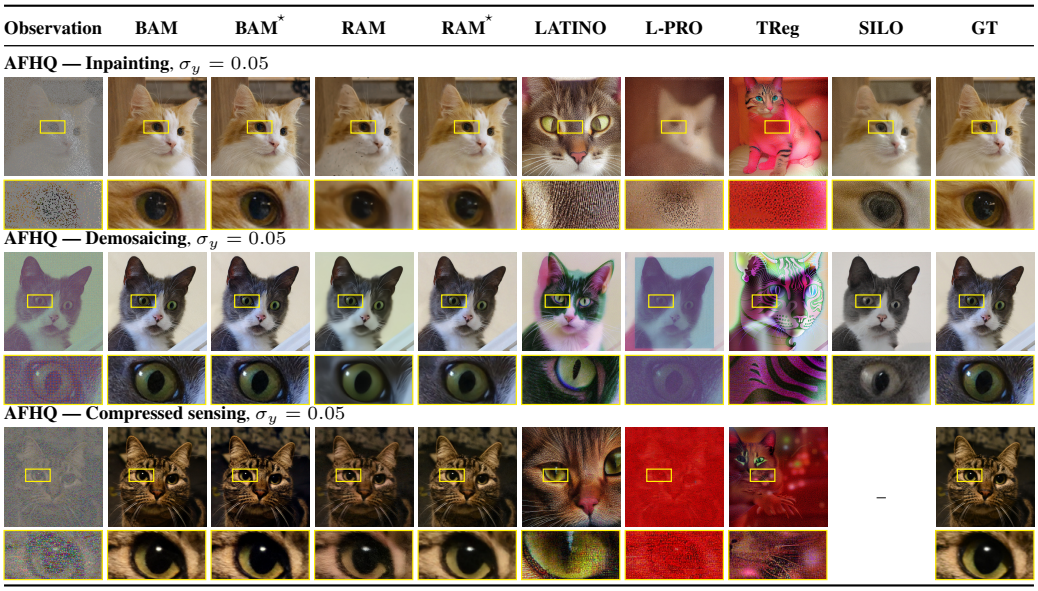

Figure 6: **AFHQ restorations** at $\sigma_y = 0.05$: inpainting, demosaicing, compressed sensing. BAM and BAM* use three steps. Strips show $4\times$ zoom. For compressed sensing, the observation column shows $A^\dagger y$; – marks unavailable results. More examples in Figure 15.

methodological side, the priorities are formal convergence and accuracy guarantees, non-Gaussian noise models, nonlinear and partially unknown forward models, larger operator libraries, automatic out-of-distribution detection, and more sophisticated backbones. We will release code and pretrained weights upon acceptance. The community can then fine-tune BAM instead of training from scratch, widening access and reducing the economic and environmental cost of deep learning for imaging.

6 AI USE DISCLOSURE

In this work, we have not used generative AI tools to generate synthetic data sets, to design or provide feedback on research methodology, to help develop theoretical models or conceptual frameworks, to implement methods, to interpret results, to support qualitative and thematic data analysis, or to assist with translation, clean and reformat dataset. Moreover, the use of generative AI tools to formulate mathematical claims, to provide critical ingredients for proving mathematical claims, to propose or refine hypotheses, or to help in the writing of proofs, are not applicable to this work. We have used generative AI tools to implement experiments and to polish writing and draft sections of the paper, notably in the appendix. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

7 REPRODUCIBILITY STATEMENT

We report in section 3, section 4, section B and section C all the main ingredients required to reproduce training and sampling of our method. We will release the code and weights upon acceptance.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 126–135, 2017. PDF.

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. arXiv:2303.08797. PDF.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.18825. PDF.

Guillermo Carbajal, Patricia Vitoria, José Lezama, and Pablo Musé. Blind motion deblurring with pixel-wise kernel estimation via kernel prediction networks. *IEEE Transactions on Computational Imaging*, 9:928–943, 2023. doi:10.1109/TCI.2023.3322012. arXiv:2308.02947.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8188–8197, 2020. arXiv:1912.01865.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. arXiv:2209.14687.

Noam Elata, Hyungjin Chung, Jong Chul Ye, Tomer Michaeli, and Michael Elad. InvFusion: Bridging supervised and zero-shot diffusion for inverse problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2504.01689.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025. PDF.

Tomer Garber and Tom Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 2398–2407, June 2025. arXiv:2412.20596.

Davis Gilton, Gregory Ongie, and Rebecca Willett. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133, 2021. doi:10.1109/TCI.2021.3118944. arXiv:2102.07944.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017. arXiv:1706.08500.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. arXiv:2006.11239.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking FID: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9307–9315, 2024. arXiv:2401.09603.

Ulugbek S. Kamilov, Charles A. Bouman, Gregery T. Buzzard, and Brendt Wohlberg. Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 40(1):85–97, 2023. doi:10.1109/MSP.2022.3199595. arXiv:2203.17061.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2019. arXiv:1812.04948.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23593–23606, 2022. arXiv:2201.11793.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Jeongsol Kim, Geon Yeong Park, Hyungjin Chung, and Jong Chul Ye. Regularization by texts for latent diffusion inverse solvers. In *The Thirteenth International Conference on Learning Representations*, 2025. arXiv:2311.15658. PDF.

Rolf Köhler, Michael Hirsch, Betty Mohler, Bernhard Schölkopf, and Stefan Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *European Conference on Computer Vision*, pp. 27–40. Springer, 2012.

Rémi Laumont, Valentin De Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, and Marcelo Pereyra. Bayesian imaging using plug & play priors: When Langevin meets Tweedie. *SIAM Journal on Imaging Sciences*, 15(2):701–737, 2022. doi:10.1137/21M1406349. arXiv:2103.04715.

Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. LSDIR: A large scale dataset for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1775–1787, 2023.

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 (ICLR)*, 2023. arXiv:2210.02747. PDF.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I2sb: Image-to-image schrödinger bridge. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22042–22062. PMLR, 2023a. arXiv:2302.05872.

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 (ICLR)*, 2023b. arXiv:2209.03003. PDF.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023. arXiv:2310.04378.

Charlesquin Kemajou Mbakam, Jonathan Spence, and Marcelo Pereyra. Learning few-step posterior samplers by unfolding and distillation of diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. arXiv:2507.02686. PDF.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021. doi:10.1109/MSP.2020.3016905. arXiv:1912.10557.

Maxence Noble, Gonzalo Iñaki Quintana, Benjamin Aubin, and Clément Chadebec. Fast, faithful and photorealistic diffusion-based image super-resolution with enhanced flow map models, 2026. arXiv:2601.16660.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. arXiv:2307.01952.

Ron Raphaeli, Sean Man, and Michael Elad. SILO: Solving inverse problems with latent operators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10570–10580, October 2025. arXiv:2501.11746. PDF.

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. doi:10.1137/16M1102884. arXiv:1611.02862.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022. arXiv:2112.10752.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.14189.

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 (ICLR)*, 2021. arXiv:2011.13456.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. arXiv:2303.01469. PDF.

Alessio Spagnoletti, Jean Prost, Andrés Almansa, Nicolas Papadakis, and Marcelo Pereyra. Latino-pro: Latent consistency inverse solver with prompt optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19597–19607, October 2025. arXiv:2503.12615.

Alessio Spagnoletti, Tim Y. J. Wang, Marcelo Pereyra, and O. Deniz Akyildiz. Consistency regularised gradient flows for inverse problems, 2026. arXiv:2605.07907.

Julián Tachella, Matthieu Terris, Samuel Hurault, Andrew Wang, Dongdong Chen, Minh-Hai Nguyen, Maxime Song, Thomas Davies, Leo Davy, Jonathan Dong, Paul Escande, Johannes Hertrich, Zhiyuan Hu, Tobías I. Liaudat, Nils Laurent, Brett Levac, Mathurin Massias, Thomas Moreau, Thibaut Modrzyk, Brayan Monroy, Sebastian Neumayer, Jérémy Scanvic, Florian Sarron, Victor Sechaud, Georg Schramm, Romain Vo, and Pierre Weiss. Deepinverse: A python package for solving imaging inverse problems with deep learning. *Journal of Open Source Software*, 10(115): 8923, 2025. doi:10.21105/joss.08923. arXiv:2505.20160.

Matthieu Terris, Samuel Hurault, Maxime Song, and Julián Tachella. Reconstruct anything model: a lightweight foundation model for computational imaging. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2503.08915.

Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-Play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948. IEEE, dec 2013. ISBN 978-1-4799-0248-4. doi:10.1109/GlobalSIP.2013.6737048. PDF.

Qinhong Yang, Dongdong Chen, Zhentao Tan, Qiankun Liu, Qi Chu, Jianmin Bao, Lu Yuan, Gang Hua, and Nenghai Yu. HQ-50K: A large-scale, high-quality dataset for image restoration. *arXiv preprint arXiv:2306.05390*, 2023.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. arXiv:1801.03924.

Jiankun Zhao, Bowen Song, and Liyue Shen. CoSIGN: Few-step guidance of ConSIstency model to solve general INverse problems. In *Computer Vision – ECCV 2024*, volume 15106 of *Lecture Notes in Computer Science*, pp. 108–126. Springer, 2024. doi:10.1007/978-3-031-73195-2_7. arXiv:2407.12676.

Zihao Zhu, Kuan-Ru Huang, Zhaoming Xu, Renjie Li, Bo Wu, Ruizheng Bai, Mingyang Wu, Sayak Paul, and Zhengzhong Tu. 4KLSDB: A large-scale dataset for 4K image restoration and text-to-image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2026. arXiv:2605.24762.