Solving Inverse Problems with FLAIR

Julius Erbach[1,2] **Dominik Narnhofer**[1] **Andreas Dombos**[1]
Bernt Schiele[2] **Jan Eric Lenssen**[2] **Konrad Schindler**[1]

[1]ETH Zürich [2]Max Planck Institute for Informatics, Saarland Informatics Campus

Abstract

Flow-based latent generative models such as Stable Diffusion 3 are able to generate images with remarkable quality, even enabling photorealistic text-to-image generation. Their impressive performance suggests that these models should also constitute powerful priors for inverse imaging problems, but that approach has not yet led to comparable fidelity. There are several key obstacles: (i) the data likelihood term is usually intractable; (ii) learned generative models cannot be directly conditioned on the distorted observations, leading to conflicting objectives between data likelihood and prior; and (iii) the reconstructions can deviate from the observed data. We present FLAIR, a novel, training-free variational framework that leverages flow-based generative models as prior for inverse problems. To that end, we introduce a variational objective for flow matching that is agnostic to the type of degradation, and combine it with deterministic trajectory adjustments to guide the prior towards regions which are more likely under the posterior. To enforce exact consistency with the observed data, we decouple the optimization of the data fidelity and regularization terms. Moreover, we introduce a time-dependent calibration scheme in which the strength of the regularization is modulated according to off-line accuracy estimates. Results on standard imaging benchmarks demonstrate that FLAIR consistently outperforms existing diffusion- and flow-based methods in terms of reconstruction quality and sample diversity. Source code is available at `https://inverseflair.github.io/`.

1 Introduction

Flow-based generative models are at the core of modern image generators like Stable Diffusion or FLUX [14]. Beyond image generation based on text prompts, these models have emerged as powerful data-driven priors for a whole range of visual computing tasks. Their comprehensive representation of the visual world, learned from internet-scale training datasets, makes them an attractive alternative to traditional handcrafted image priors. Often, they can be used without any task-specific retraining.

While it is evident that a model capable of generating photorealistic images should be suitable as prior (a.k.a. regularizer) for inverse imaging problems, a practical implementation faces several challenges. On the one hand, flow-based models normally operate in the lower-dimensional latent space of a variational autoencoder (VAE), which means that the forward operator (the relationship between the observed, degraded image and the desired, clean target image) is no longer linear. On the other hand, the iterative nature of the generative process means that intermediate stages are corrupted with (time-dependent) random noise. Hence, one cannot explicitly evaluate their data likelihood, which renders the data term intractable. Moreover, learned generative models tend to overly favor regions of the training distribution that have a high sample density. For test samples that fall in low-density regions, the prior will have a too strong tendency to pull towards outputs with higher a-priori likelihood, compromising fidelity to the input observations.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

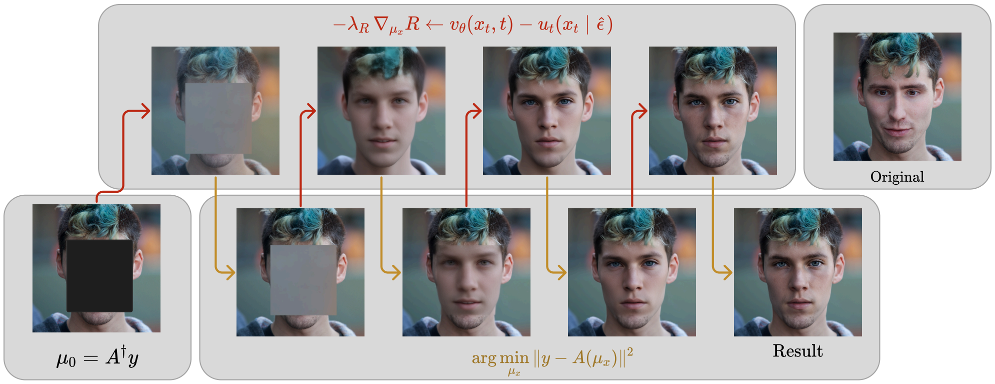

$-\lambda_R \nabla_{\mu_x} R \leftarrow v_\theta(x_t, t) - u_t(x_t \mid \hat{\epsilon})$
Original
$\mu_0 = A^\dagger y$
$\arg\min_{\mu_x} \|y - A(\mu_x)\|^2$
Result

Figure 1: Starting from the adjoint based initialization, we alternate between (i) regularizer updates via a flow-matching loss that aligns the velocity u_t of the variational distribution with the learned velocity field v_θ, and (ii) hard data consistency steps that project the current estimate onto the measurement manifold.

Here, we propose flow-based latent adaptive inference with deterministic re-noising (FLAIR), a novel, training-free variational framework explicitly tailored to integrate flow-based latent diffusion models into inverse problem-solving. To the best of our knowledge, FLAIR is the first scheme that combines latent generative modeling, flow matching and variational inference into a unified formulation for inverse problems. Our main contributions are

- A novel variational objective for inverse problems with flow-matching priors.
- Deterministic trajectory adjustments guide the prior towards regions which are more consistent with the observed data.
- Decoupled optimization of data and regularization terms, enabling hard data consistency.
- A novel, time-dependent weighting scheme, calibrated via offline accuracy estimates, that adapts the regularization along the flow trajectory to match the changing reliability of the model's predictions, ensuring robust inference.

2 Related Work

Deep learning based priors. Deep learning–based methods typically follow one of two main approaches: they either directly learn an inverse mapping [28, 18, 30, 4, 60], or aim to learn a suitable prior, either through non-generative approaches like unrolled optimization networks [15, 29, 1, 36] or through generative models such as generative adversarial networks [37, 7, 46], or diffusion [17, 49, 52] respectively flow based models [31, 33]. The latter have demonstrated impressive performance in image generation tasks, sparking growing interest in leveraging them as priors for solving inverse problems, particularly through posterior sampling techniques.

Posterior sampling. Although incorporating the implicit prior learned from a denoising-based deep neural network seems straightforward, distinct challenges arise depending on the framework. While early stochastic approaches utilized blind universal denoisers to solve inverse problems [21], they rely on heuristic scheduling and are limited to linear operators. Modern diffusion models formalize this process via fixed forward trajectories, yet this time-dependent structure renders the likelihood intractable [11] A variety of approaches have been proposed for diffusion-based posterior sampling [58, 19, 35, 20]: enforcing the trajectory to stay on the respective noise manifold [11, 12, 61], applying an SVD to run diffusion in the spectral domain [25], utilizing range-null space decomposition during the reverse diffusion process [25], guidance by the pseudo-inverse of the forward operator [50].

Many prior methods perform well in pixel space but are difficult to apply in latent diffusion models due to VAE non-linearity or memory constraints. In order to circumvent this issue, the authors of

ReSample [48] rely on enforcing hard data consistency through optimization and resampling during the reverse diffusion process. PSLD [42], introduces additional objectives terms to ensure that all gradient updates point to the same optima in the latent space. FlowChef [39] incorporates guidance into the flow trajectory during inference, whereas FlowDPS [26] separates the update step into two components: one for estimating the clean image and another for estimating the noise.

In contrast FLAIR follows another class of posterior sampling-based methods, which integrate diffusion priors with inverse problems by directly optimizing a variational objective that approximates the data posterior [34]. This framework was recently extended by RSD [62], which incorporates a repulsion mechanism to promote sample diversity and applied to latent diffusion models. A known issue with this type of optimization is mode collapse [40], which leads to blurry results for these methods. Our method targets this problem by introducing a deterministic trajectory adjustment.

3 Background

3.1 Inverse problems

In many imaging tasks, such as inpainting [5], super-resolution [38] or tomographic reconstruction [47], one aims to recover a target signal $x \in \mathbb{R}^n$ from a distorted observation $y \in \mathbb{R}^m$. The observation is regarded as the result of applying a forward operator $\mathcal{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$ to the target signal, corrupted by additive Gaussian noise $\nu \in \mathbb{R}^m$ with standard deviation σ_ν.

$$y = \mathcal{A}x + \nu. \tag{1}$$

In most practical applications, the forward operator $\mathcal{A}$ is either non-invertible or severely ill-conditioned, making (1) generally ill-posed.

Variational methods solve ill-posed inverse problems by minimizing an energy functional

$$\mathcal{E}(x, y) = \mathcal{D}(x, y) + \mathcal{R}(x). \tag{2}$$

to recover the solution.

Interpreted probabilistically via Bayes' theorem, the posterior distribution $p(x|y)$ is proportional to the product $p(y|x)p(x)$. In the negative log-domain, this yields the data term $\mathcal{D}(x, y) = -\log p(y|x)$ and the regularizer $\mathcal{R}(x) = -\log p(x)$. Handcrafted priors based on regularity assumptions like sparsity [44, 45, 10, 13, 9] have long been the standard, but have largely been replaced by deep learning-based methods in modern data-driven schemes.

3.2 Flow based priors

Models based on flow matching [31] learn a time-dependent vector field $v_\theta(x_t, t)$ that continuously transforms samples from a simple initial distribution $p_1(x)$ to a complex target data distribution $p_0(x)$. Formally, this transformation is described by solving the ordinary differential equation (ODE):

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = v_{\theta,t}(\psi_t(x)), \quad t \in [0, 1], \tag{3}$$

where $\psi_t(x)$ represents the trajectory of a sample, evolving smoothly from an initial value drawn at $t = 1$ toward a target value at $t = 0$.

Since the integrated ODE path maps the simple distribution $p_1(x)$ to the complex target $p_0(x)$, the learned flow-based model captures the structure of the data and can therefore serve as a powerful prior for solving inverse problems. To make this approach tractable for high-resolution data, we adopt the latent diffusion model (LDM) framework [41], which shifts the generative process to a lower-dimensional latent space using a pretrained autoencoder with encoder $E : \mathbb{R}^n \mapsto \mathbb{R}^d$ and decoder $D : \mathbb{R}^d \mapsto \mathbb{R}^n$, where $d \ll n$. However, applying such priors to inverse problems introduces challenges, as the non-linearity of the VAE disrupts the linear relationship between measurements and the target signal, resulting in a nonlinear forward operator.

3.3 Variational flow sampling

To solve inverse problems from a Bayesian perspective, we aim to sample from the posterior

$$p(x_0|y) \propto p(y|x_0)p(x_0), \tag{4}$$

where the likelihood is given by $p(y|x_0) = \mathcal{N}(\mathcal{A}x_0, \sigma^2 \mathrm{Id})$, and $p(x_0)$ represents the prior modeled by the flow-based generative model.

Inspired by previous work [34, 62] we introduce a variational distribution $q(x_0|y) = \mathcal{N}(\mu_x, \sigma_x^2)$ to approximate the true posterior $p(x_0|y)$, by minimizing their Kullback–Leibler divergence:

$$q(x_0|y) \in \arg \min_{q(x_0|y)} \mathrm{KL}(q(x_0|y) \| p(x_0|y)). \tag{5}$$

Rewriting the KL divergence by means of the variational lower bound leads to:

$$KL(q(x_0|y)\|p(x_0|y)) = -\underbrace{\mathbb{E}_{q(x_0|y)}[\log p(y|x_0)]}_{\mathcal{D}(x,y)} + \underbrace{KL(q(x_0|y)\|p(x_0))}_{\mathcal{R}(x)} + \underbrace{\log p(y)}_{\text{const}}. \tag{6}$$

Since a single Gaussian cannot capture a multi-modal posterior, we simplify to a deterministic approximation, setting $\sigma_x^2 = 0$. Equivalently, this corresponds to a single-particle approximation in the sense of Stein variational methods [32]. As shown in [51], rewriting Equation 6 under this approximation and extending it to the time-dependent noisy posterior yields:

$$\arg \min_{q(x_0|y)} \underbrace{\mathbb{E}_{q(x_0|y)} \left[\frac{\|y - f(\mu_x)\|^2}{2\sigma_\nu^2} \right]}_{\mathcal{D}(x,y)} + \underbrace{\int_0^T \omega(t)\, \mathbb{E}_{q(x_t|y)} \left[\|\nabla_x \log q(x_t|y) - \nabla_x \log p(x_t)\|^2 \right] dt}_{\mathcal{R}(x)} \tag{7}$$

The first term in Equation 7 describes the data term $\mathcal{D}(x,y)$ and the second the regularizer $\mathcal{R}(x)$, where the integral ensures optimization over the entire diffusion trajectory. Notably, the latter constitutes a weighted score-matching objective, where $\nabla_x \log p(x_t)$ represents the score function [52], which may be extracted from a pretrained diffusion or flow model.

The score of the noisy variational distribution depends on the forward diffusion process and can be computed analytically.

Note that for $\omega(t) = \beta(t)/2$ the weighted score-matching loss recovers the gradient of the diffusion model's evidence lower bound, so that optimizing it yields the maximum likelihood estimate of the data distribution [51]. However, optimizing Equation 7 is costly, as it requires computing the gradient through the flow model. As shown in [54] this can be circumvented by reformulating the regularizer in terms of the Wasserstein gradient flow:

$$\nabla_{\mu_x} \mathcal{R}(x) = \mathbb{E}_{t, q(x_t|y)} \left[\omega(t) (\underbrace{\nabla_x \log q(x_t|y)}_{\text{score of noisy variational distribution}} - \underbrace{\nabla_x \log p(x_t)}_{\text{score of noisy prior distribution}}) \right] \tag{8}$$

Note that optimizing only the regularization term, without the data term, at test time is equivalent to the objective of Score Distillation Sampling (SDS) [40].

4 Method

Flow Formulation. The variational formulation in Equation 7 is formulated for the score, but can be reformulated into a denoising or ϵ_θ parameterization [52, 34]. However, we are interested in a variational objective that depends on the velocity field $v_\theta(x_t, t)$, which characterizes the probabilistic trajectory that connects the noise and data distributions.

Proposition 1. *We propose to replace the score-based regularizer in the standard variational objective with a flow matching formulation, resulting in the following objective function:*

$$\arg \min_{q(x_0|y)} \underbrace{\mathbb{E}_{q(x_0|y)} \left[\frac{\|y - f(\mu_x)\|^2}{2\sigma_\nu^2} \right]}_{\mathcal{D}(x,y)} + \underbrace{\int_0^T \lambda_{\mathcal{R}}(t)\, \mathbb{E}_{q(x_t|y)} \left[\|v_\theta(x_t, t) - u_t(x_t|\epsilon)\|^2 \right] dt}_{\mathcal{R}(x)} \tag{9}$$

$$\nabla_{\mu_x} \mathcal{R}(x) = \mathbb{E}_{t, q(x_t|y)} \left[\lambda_{\mathcal{R}}(t) v_\theta(x_t, t) - u_t(x_t \mid \epsilon) \right] \tag{10}$$

The flow-matching term that defines the regularizer arises by reparameterizing the variational distribution to $q(x_t|y) = \mathcal{N}((1-t)\mu_x, t^2\, I)$. This corresponds to sampling via the deterministic map $\psi_t(x_0 \mid \epsilon) = (1-t)\, x_0 + t\, \epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$. By reformulating the score in terms of the target velocity field u_t, we get:

$$\nabla_x \log q(x_t|y) = -\frac{(1-t)u_t(x_t|\epsilon) + x_t}{t} \tag{11}$$

For the learned velocity $v_\theta(x_t, t)$ a similar approximation holds – for a full derivation, see the supplementary material subsection A.3.

$$v_\theta(x_t, t) \approx \frac{-t\nabla_x \log p(x_t) - x_t}{1-t} \tag{12}$$

We can therefore approximate the score of the noisy prior with our learned velocity field v_θ

$$\nabla_x \log p(x_t) \approx -\frac{(1-t)v_\theta(x_t, t) + x_t}{t}. \tag{13}$$

Hard Data Consistency. Existing variational posterior sampling approaches [34, 62] impose soft constraints on the data fidelity term $\mathcal{D}(x, y)$. In contrast, recent work [48] has demonstrated that, when sampling from latent diffusion models, enforcing hard data consistency generally leads to better reconstructions with improved visual fidelity. Our method shares this motivation, but differs in that we optimize over a variational distribution, i.e., we compute $\min\ \mathbb{E}_{q(x_0|y)}[-\log p(y|x_0)]$. An additional advantage of this variational setup is that it allows us to initialize the optimization variable with an adjoint based initialization $\mu_x = E(A^\top y)$, with E being the encoder of the VAE and $A^\top$ the adjoint of the linear forward operator in pixel space. Other initialization strategies are also possible.

Accuracy Calibration. As our framework evaluates the trajectory at each time step, we aim to weight the regularizer's contribution according to its reliability. The difficulty of the prediction task has been shown to depend on the network parameterization, as well as on the specific time step t [22]. Since the regularization term $\mathcal{R}(x)$ in our approach is equivalent to the training objective of the pre-trained flow model, we can easily weight it by the expected model error, which we calibrate on a small set of images. Specifically, we sample N calibration images and compute the conditional flow matching objective for 100 linearly spaced time steps between 0 and 1, then average the error over all images to obtain the expected model error at each time step. Different functions of the model error can be chosen as weight for the regularizer. We choose:

$$\lambda_{\mathcal{R}}(t) = \frac{1}{N}\left(\sum_{i=1}^{N} \left\| v_\theta(x_t^{(i)}, t) - u_t(x_t^{(i)} \mid \epsilon) \right\|^2\right)^{-1} \tag{14}$$

and set $\lambda_{\mathcal{R}}(t) = 0$ for all $t < 0.2$, since the accuracy of SD3 is heavily degraded for low noise levels.

Deterministic Trajectory Adjustment. Score distillation sampling relies on the assumption that $x_t = (1-t)\mu_x + t\epsilon$ lies in a region of the learned prior that has reasonably high support/density. In practice, this is not always the case. When not tightly conditioned (usually with extensive text prompts), even the best available diffusion models assign low density to many plausible regions of the latent space, leading to bad gradient steps. Therefore, we increase the probability of $p(x_t)$ by additionally conditioning x_t on the estimated "end-point" μ.

Proposition 2. *We introduce a reparameterized variational distribution with a mean that linearly interpolates between the posterior mean μ_x and a model-guided $\hat{x}_1$:*

$$q(x_t \mid y) = \mathcal{N}\left((1-t)\mu_x + t\alpha\hat{x}_1,\ t^2(1-\alpha^2)I\right), \tag{15}$$

where $\hat{x}_1 = x_{t+\delta t} + (1-t-\delta t)v_\theta(x_{t+\delta t}, t+\delta t)$ is a single-step velocity-based predictor, and $\alpha \in [0,1]$ controls the trade-off between deterministic guidance and random noise. This reparameterization induces the following reference velocity field:

$$u_t(x_t \mid \epsilon) = \frac{\alpha\hat{x}_1 + \sqrt{1-\alpha^2}\epsilon - x_t}{1-t}. \tag{16}$$

Intuitively, changing the formulation in this manner ensures that the model relocates the sample to its expected position on the learned manifold rather than injecting arbitrary noise, which could drive it in a direction that has high prior likelihood but is not consistent with the observation. To further encourage exploration and avoid collapsing onto the trajectory of the adjoint measurement, we inject an additional stochastic component ϵ during this process. A full derivation can be found in the supplementary material, subsection A.3.

4.1 Algorithm

The following pseudo-code summarizes our method, integrating all the components discussed above.

We adapt the standard scheme [34, 62] of linearly traversing time in a descending manner and stop at $t = 0.2$ as explained in section 4. We choose $\alpha = 1 - t$. Gradient updates to enforce hard data consistency are performed using stochastic gradient descent. For further implementation details and ablations, see subsection A.4

Algorithm 1: The FLAIR solver for inverse imaging problems

```
Input: $\mu_x = \mu_{init}$, $\lambda_R$, $\alpha$, $y$, $\mathcal{A}$, $v_\theta$
Output: $\mu_x$
$\hat{\epsilon} \sim \mathcal{N}(0, I)$;                                  ▷ initial noise sample
for $t \leftarrow 1$ to $0$ by $-\Delta t$ do
    $x_t \leftarrow (1-t)\,\mu_x + t\,\hat{\epsilon}$;                         ▷ sample noisy latent
    $u_t(x_t \mid \hat{\epsilon}) \leftarrow \frac{\hat{\epsilon} - x_t}{1-t}$;
    $\nabla_{\mu_x} R \leftarrow v_\theta(x_t, t) - u_t(x_t \mid \hat{\epsilon})$;
    $\mu_x \leftarrow \mu_x - \lambda_R \nabla_{\mu_x} R$;                      ▷ update w.r.t. regularizer
    $\mu_x \leftarrow \arg\min_{\mu_x} \|y - \mathcal{A}(\mu_x)\|^2$;              ▷ hard data consistency
    $\epsilon \sim \mathcal{N}(0, I)$;
    $\hat{x}_1 \leftarrow x_t + (1-t)\,v_\theta(x_t, t)$;                    ▷ predict deterministic noise
    $\hat{\epsilon} \leftarrow \alpha\,\hat{x}_1 + \sqrt{1-\alpha^2}\,\epsilon$;                   ▷ update noise estimate
```

5 Experiments

We evaluate the performance of FLAIR in a variety of inverse imaging tasks and compare it against several baselines, using the SD3 backbone without any fine-tuning. We used several metrics including SSIM [55], LPIPS [59] and patchwise FID [16] (pFID) to comprehensively assess the perceptual and quantitative quality of the reconstructions. FID is computed using InceptionV3 features on patches of 256x256 resolution. All experiments were performed on a NVidia RTX 4090 GPU with 24GB of VRAM. For completeness we also show PSNR values, but point out that the metric is not well suited for our purposes: PSNR favors the posterior mean, while the goal of the variational approach is to sample from the posterior distribution. Accordingly, PSNR is known to prefer over-smoothed, blurry outputs over sharp ones [6]. To demonstrate that our model can also produce accurate MMSE estimates, we performed ensemble predictions by running posterior sampling eight times and averaging the results. As shown in subsection A.11, ensembling improves PSNR values while reducing LPIPS. This confirms that our samples are distributed around the posterior mean. Moreover, it shows that results closer to the posterior mean – such as those produced by baseline methods – are perceptually farther from the ground truth (in LPIPS) compared to our samples.

5.1 Setup

Datasets. We utilize two high-resolution image datasets: FFHQ [23] and DIV2K [2]. FFHQ consists of 70k diverse face images at 1024×1024 resolution of which we take the first 1000 samples. It is covering variations in age, pose, lighting, and ethnicity. DIV2K contains 800 high-quality images in 2K resolution that span a range of natural scenes with varied textures and structures.

Baselines. Our method is benchmarked against several recent inverse imaging solvers based on posterior sampling. Specifically, we compare to ReSample [48], FlowDPS [26], FlowChef [39], and RSD [62]. The latter is used without repulsive term as it delivers better results. To ensure a fair and meaningful comparison, all methods are evaluated with the same number of function evaluations.

Problem Setting. We run and evaluate all methods at a fixed output resolution of 768×768 pixels. For single image super-resolution, we consider scaling factors of 8× and 12×. The corresponding low-resolution inputs are generated by bicubic downsampling. Motion blur is simulated with a blur kernel of size 61. For box inpainting, we mask large, continuous rectangles that cover approximately one third of the observation. All synthesized observations are corrupted with additive Gaussian noise, with standard deviation σ_ν of 0.5%.

For inference on the FFHQ dataset, we use a predefined text prompt of the form *"A high quality photo of a face"*, and for DIV2k *"A high quality photo of"* concatenated with an image-specific description retrieved by applying DAPE [56] to the observation.

5.2 Experimental Results

Inverse Problems. Our experiments clearly demonstrate that FLAIR outperforms existing flow-based approaches in terms of all perceptual metrics, see Table 1.

In the case of image inpainting, our method produces high-quality reconstructions that fully leverage the power of the generative model and blend naturally into the surrounding context, avoiding degradations and artifacts that we observe in the baselines. In particular, FlowDPS tends to produce implausible textures in the inpainted regions, while FlowChef regularly fails to generate semantically consistent content at all.

For single-image super-resolution, FLAIR consistently delivers the most perceptually convincing and realistic outputs. Notably, the FID scores remain low for both ×8 and ×12 magnification, indicating an effective usage of the generative prior to overcome the increasing ill-posedness. Again, FlowDPS suffers from blur and low texture quality, whereas FlowChef tends to lose semantic coherence.

In motion deblurring, FLAIR also restores sharper and semantically more credible content than competing approaches, which often suffer from residual blur or inconsistent details. The boost in reconstruction quality is quantitatively reflected by all metrics, confirming that FLAIR reconstructs images with high fidelity. For further qualitative examples, see subsection A.13.

Table 1: **Quantitative results** with 50 NFE and $\sigma_\nu = 0.5\%$.

	SR ×8				SR ×12				Motion Deblurring				Inpainting			
Method	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑
FFHQ 768×768																
ReSample	0.400	55.6	**0.815**	26.37	0.474	80.3	**0.786**	25.47	0.457	82.9	**0.788**	25.45	0.366	70.8	0.827	21.83
FlowDPS	0.374	38.5	0.756	29.24	0.413	44.0	0.741	28.05	0.431	54.3	0.732	27.64	0.344	42.5	0.771	19.19
RSD	0.391	51.7	0.776	**29.69**	0.462	71.7	0.743	**28.11**	0.458	77.3	0.743	27.67	0.478	73.3	0.736	21.97
FlowChef	0.341	30.5	0.760	28.42	0.373	46.5	0.730	27.00	0.406	40.2	0.716	25.81	0.394	69.8	0.780	18.18
Ours	**0.213**	**13.3**	0.777	29.54	**0.271**	**16.2**	0.740	27.71	**0.236**	**10.7**	0.772	**29.61**	**0.184**	**8.7**	**0.828**	**23.69**
DIV2K 768×768																
ReSample	0.533	55.0	**0.625**	22.34	0.643	88.1	**0.562**	20.85	0.556	79.7	0.617	21.79	0.285	51.9	0.796	22.68
FlowDPS	0.476	44.4	0.567	23.01	0.547	54.0	0.528	21.79	0.558	65.5	0.536	21.88	0.328	29.2	0.692	21.71
RSD	0.539	60.9	0.591	**23.45**	0.684	95.7	0.523	**21.96**	0.638	97.6	0.551	22.10	0.464	63.9	0.678	23.23
FlowChef	0.490	36.5	0.539	21.84	0.525	43.8	0.492	20.52	0.561	49.6	0.486	19.90	0.489	58.3	0.659	20.87
Ours	**0.353**	**26.5**	0.607	23.30	**0.421**	**32.1**	0.525	21.39	**0.315**	**21.1**	**0.653**	**24.44**	**0.163**	**11.0**	**0.815**	**23.75**

Posterior Variance. To demonstrate that FLAIR does not suffer from mode collapse, we assess the posterior variance $\mathrm{Var}[x|y]$ for the task of ×12 Super Resolution, by drawing 32 samples for a fixed observation y and computing their pixel-wise variance. We conduct that experiment for our FLAIR approach, for RSD with repulsive term, and for FlowDPS [26]. The example in Figure 3 illustrates that FLAIR has the highest sample diversity, which is also reflected in the corresponding variance maps. Notably, the sample variance is concentrated in regions with high-frequency textures. This indicates that our method reliably reconstructs the posterior, whose low-frequency part is, in the super-resolution setting, tightly constrained by the likelihood term.

Editing. Beyond image restoration, we observe that our method also performs remarkably well for text-based image editing, simply presenting suitable target prompts during inpainting. Figure 4 illustrates a variety of edited images generated from the same photograph with the help of the depicted masks and prompts.

Pixel Space Experiments We also implement FLAIR in pixel-space using the model from [33], trained on CelebA-HQ resized to 256x256 px. We compare to DDNM [53], DPS [11], Moment Matching [43] and ΠGDM [50]. We tuned the hyperparameters for all baselines, which we report in subsection A.9. As shown in Table 5.2, our method also outperforms previous work in the pixel space, demonstrating its broader applicability.

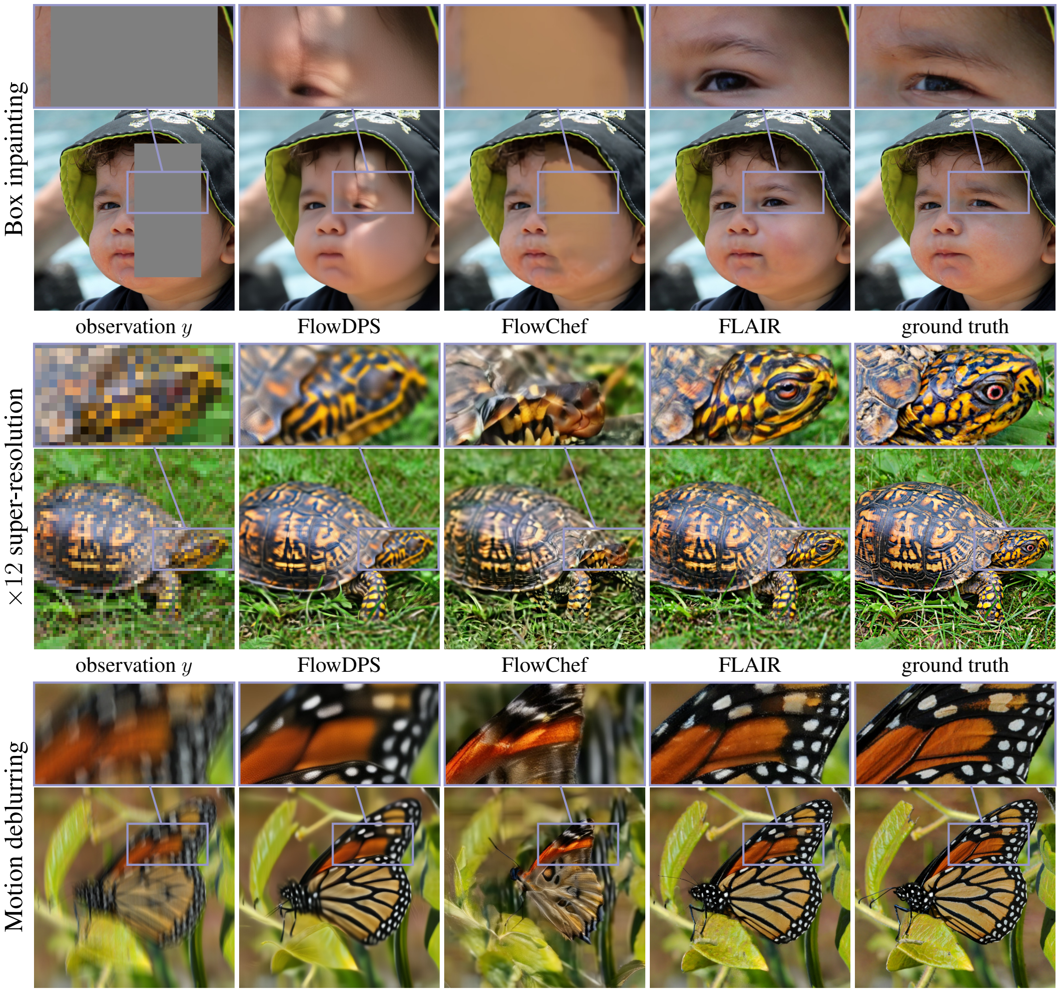


Figure 2: **Qualitative comparison**. FLAIR produces posterior samples of high perceptual quality while maintaining high data likelihood. Best viewed zoomed in.

Table 2: **Quantitative results** with 50 NFE and $\sigma_\nu = 0.5\%$ – In-painting and Super-resolution ($\times 8$).

	Inpainting				SR ×8			
Method	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑
DDNM	0.158 ± 0.042	26.9	0.732 ± 0.037	18.31 ± 2.94	0.199 ± 0.052	31.9	0.635 ± 0.079	23.59 ± 1.64
DPS	0.195 ± 0.064	30.2	0.689 ± 0.077	20.49 ± 2.81	0.172 ± 0.058	27.8	0.658 ± 0.088	24.59 ± 2.04
MM	0.161 ± 0.054	28.8	0.728 ± 0.062	20.59 ± 3.37	0.172 ± 0.051	29.1	0.669 ± 0.083	24.65 ± 1.97
ΠGDM	0.195 ± 0.064	30.2	0.689 ± 0.077	20.49 ± 2.81	0.157 ± 0.052	26.5	0.677 ± 0.084	24.98 ± 2.07
FLAIR	$\mathbf{0.097} \pm \mathbf{0.035}$	**14.2**	$\mathbf{0.831} \pm \mathbf{0.031}$	$\mathbf{21.87} \pm \mathbf{2.66}$	$\mathbf{0.143} \pm \mathbf{0.039}$	**22.9**	$\mathbf{0.712} \pm \mathbf{0.076}$	$\mathbf{25.93} \pm \mathbf{1.96}$

5.3 Ablation Studies

We systematically analyze the impact of key design choices in our method. Specifically, we ablate the deterministic trajectory adjustment, the use of hard data consistency, and the calibration of the regularizer weight for ×12 super-resolution, using a subset of 100 samples from the FFHQ and DIV2K datasets. Quantitative and qualitative results are shown in Table 3 and Figure 6, respectively.

Hard Data Consistency (HDC). Dropping the hard data consistency degrades both metrics, with PSNR being particularly affected due to poorer alignment with the input observation, which is also evident in the visual example: the reconstruction is plausible but deviates from the observation.

Deterministic Trajectory Adjustment (DTA). The biggest performance drop compared to the full setup occurs when removing the deterministic trajectory adjustment, as random noise sampling harms

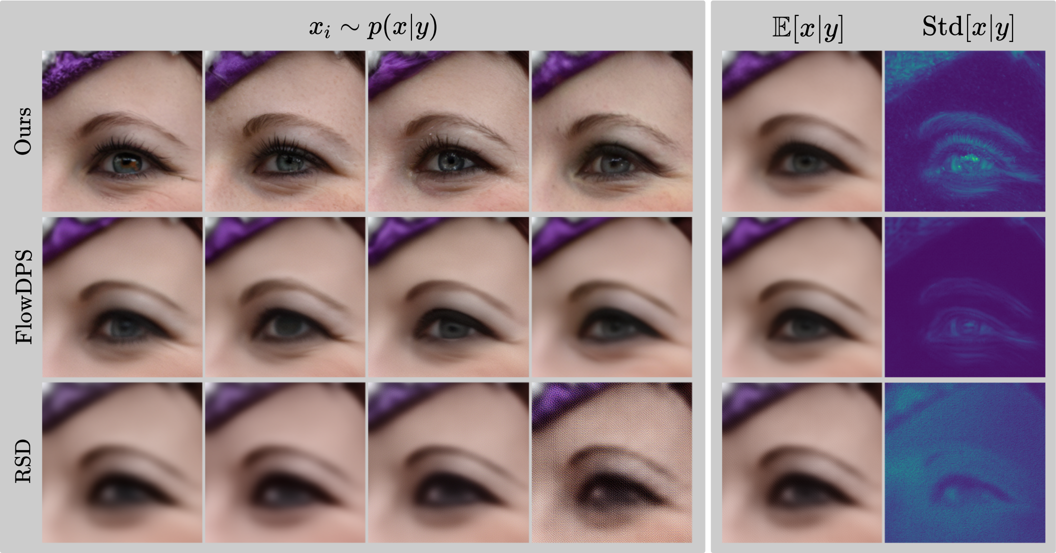
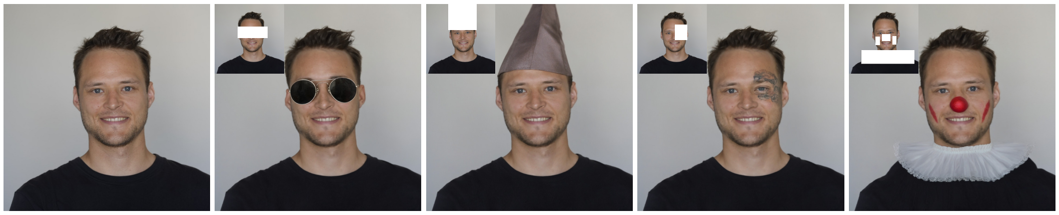


Figure 3: **Zoomed-in reconstructions for x12 Super Resolution.** We show posterior samples (col. 1–4) of FLAIR, FlowDPS, and RSD, posterior mean and standard deviation (over 32 samples, col. 5,6). 0 ▬ 0.16



Original image

"man wearing aviator glasses."

"man wearing a wizard hat"

"man with a Mike Tyson facial tattoo"

"clown with a red nose, red makeup and a ruff"

Figure 4: **Edited images** shown alongside original, with prompts: "A high resolution portrait of a..."

the gradient updates in low-density regions of the prior. The reconstruction appears overly smooth and lacks texture details.

Calibrated Regularizer Weight (CRW). Replacing our calibrated regularizer weight with $\lambda_{\mathcal{R}}(t) = t$ also has a strong impact on perceptual quality: the result is visibly blurred if one ignores the changing accuracy of the regularizer along the flow trajectory.

Table 3: **Ablation study** for $\times 12$ super-resolution on DIV2K and FFHQ. Model components are individually switched on or off.

HDC	DTA	CRW	FFHQ		DIV2K	
			LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑
✓	✓	✓	0.259	27.45	0.427	21.05
✗	✓	✓	0.297	27.17	0.467	20.82
✓	✗	✓	0.432	27.20	0.622	21.69
✓	✓	✗	0.363	28.58	0.583	21.98
✗	✗	✗	0.392	28.33	0.605	21.99

Legend. HDC: Hard Data Consistency; DTA: Deterministic Trajectory Adjustment;
CRW: Calibrated Regularizer Weight. ✓ = included, ✗ = ablated.

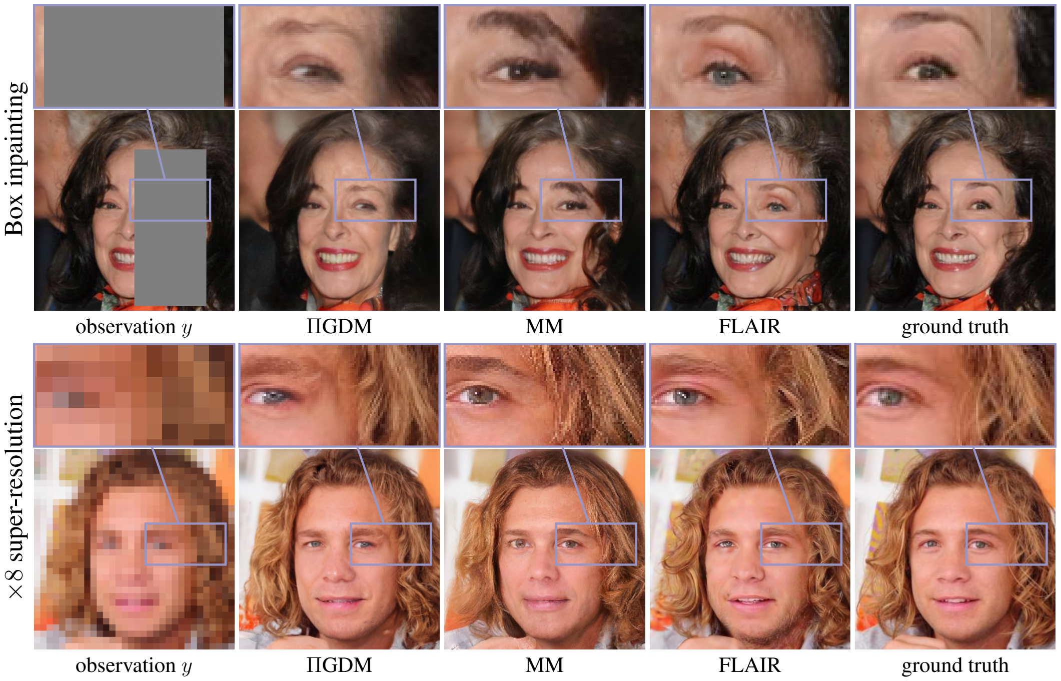


Figure 5: **Qualitative comparison.** FLAIR in pixel space produces posterior samples.

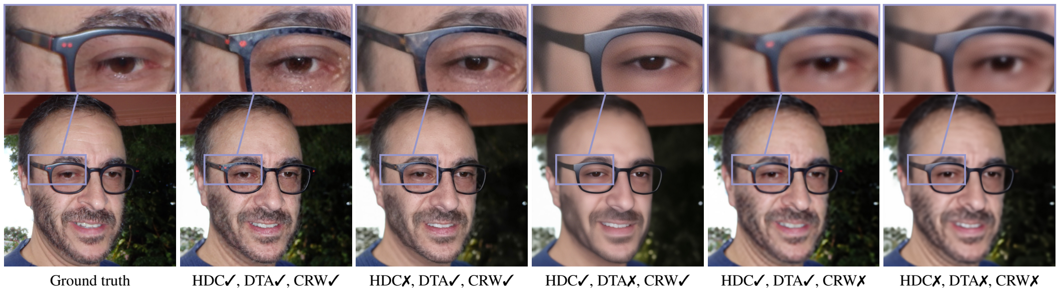


Figure 6: **Qualitative samples from the ablation study** on $\times 12$ Super Resolution.

6 Conclusion and Limitations

We have presented FLAIR, a training-free variational framework for inverse problems that uses a flow-based generative model as its image prior. By combining the power of (latent) flow-based models with a principled reconstruction of the posterior distribution, FLAIR addresses key limitations of existing methods. First, it is able to target the generation towards images, which match the observation, by aiding the degradation-agnostic flow matching loss with deterministic noise vectors. Second, it enables hard data consistency without sacrificing sample diversity, by decoupling the data consistency constraint from the regularization, while adaptively reweighting the latter according to its expected accuracy, calibrated offline. Experiments with different image datasets and tasks confirm that FLAIR consistently achieves higher reconstruction quality than existing baselines based on either flow matching or denoising diffusion. Notably, our proposed method achieves, at the same time, excellent perceptual quality, close adherence to the input observations, and high sample diversity.

Evidently, FLAIR inherits the limitations of the underlying generative model. These include biases caused by the selection of training data, constraints w.r.t. the output resolution, and a limited ability to recover out-of-distribution modes. Furthermore, our approach introduces additional hyper-parameters needed to control the deterministic trajectory adjustment. We note that high fidelity image restoration methods can potentially be misused for unethical image manipulations.

Acknowledgments This work was funded, in part, by the Max Plank ETH Center for Learning Systems and Huawei Technologies Oy (Finland) Co. Ltd.

References

[1] Hemant K Aggarwal, Merry P Mani, and Mathews Jacob. MoDL: Model-based deep learning architecture for inverse problems. *IEEE Transactions on Medical Imaging*, 38(2):394–405, 2018.

[2] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, 2017.

[3] 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*, July 2017.

[4] Alexander Becker, Rodrigo Caye Daudt, Nando Metzger, Jan Dirk Wegner, and Konrad Schindler. Neural fields with thermal activations for arbitrary-scale super-resolution. *arXiv:2311.17643*, 2023.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[6] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *CVPR*, 2018.

[7] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *ICML*, 2017.

[8] Levi Borodenko. motionblur: Generate authentic motion blur kernels and apply them to images. `https://github.com/LeviBorodenko/motionblur`, 2025. Accessed: 2025-05-19.

[9] Kristian Bredies and Martin Holler. Higher-order total variation approaches and generalisations. *Inverse Problems. Topical Review*, 36(12):123001, 2020.

[10] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[11] Hyungjin Chung, Jeongsol Kim, Michael T McCann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023.

[12] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *NeurIPS*, 35, 2022.

[13] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied*, 57(11):1413–1457, 2004.

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[15] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic Resonance in Medicine*, 79(6):3055–3071, 2018.

[16] 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 *NeurIPS*, 2017.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

[18] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. *NeurIPS*, 2008.

[19] Yazid Janati, Badr Moufad, Mehdi Abou El Qassime, Alain Oliviero Durmus, Eric Moulines, and Jimmy Olsson. A mixture-based framework for guiding diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

[20] Yazid Janati, Badr Moufad, Alain Durmus, Eric Moulines, and Jimmy Olsson. Divide-and-conquer posterior sampling for denoising diffusion priors. *Advances in Neural Information Processing Systems*, 37:97408–97444, 2024.

[21] Z Kadkhodaie and E P Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Adv. Neural Information Processing Systems (NeurIPS)*, volume 34, pages 13242–13254, (virtual), Dec 2021. Curran Associates, Inc.

[22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. Introduces the Flickr-Faces-HQ (FFHQ) dataset.

[25] Bahjat Kawar, Gregory Vaksman, and Michael Elad. SNIPS: Solving noisy inverse problems stochastically. *NeurIPS*, 34, 2021.

[26] Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. *arXiv:2503.08136*, 2025.

[27] Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. `https://https://github.com/FlowDPS-Inverse/FlowDPS`, 2025. Accessed: 2025-05-19.

[28] Florian Knoll, Jure Zbontar, Anuroop Sriram, Matthew J. Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzalv, Adriana Romero, Michael Rabbat, Pascal Vincent, James Pinkerton, Duo Wang, Nafissa Yakubova, Erich Owens, C.Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: A publicly available raw k-space and DICOM dataset of knee images for accelerated MR image reconstruction using machine learning. *Radiology: Artificial Intelligence*, 2(1):e190007, 2020.

[29] Erich Kobler, Alexander Effland, Karl Kunisch, and Thomas Pock. Total deep variation: A stable regularization method for inverse problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[30] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[31] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[32] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NeurIPS*, volume 29, pages 2378–2386, 2016.

[33] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv:2209.03003*, 2022.

[34] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *ICLR*, 2024.

[35] Badr Moufad, Yazid Janati, Lisa Bedin, Alain Durmus, Randal Douc, Eric Moulines, and Jimmy Olsson. Variational diffusion posterior sampling with midpoint guidance. *arXiv preprint arXiv:2410.09945*, 2024.

[36] Dominik Narnhofer, Alexander Effland, Erich Kobler, Kerstin Hammernik, Florian Knoll, and Thomas Pock. Bayesian uncertainty estimation of learned variational MRI reconstruction. *IEEE Transactions on Medical Imaging*, 41(2):279–291, 2021.

[37] Dominik Narnhofer, Kerstin Hammernik, Florian Knoll, and Thomas Pock. Inverse GANs for accelerated MRI reconstruction. In *Wavelets and Sparsity XVIII*, volume 11138. SPIE, 2019.

[38] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003.

[39] Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. Steering rectified flow models in the vector field for controlled image generation. *arXiv:2412.00100*, 2024.

[40] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[42] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. In *NeurIPS*, 2023.

[43] François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization, 2024.

[44] 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.

[45] Sylvain Sardy, Paul Tseng, and Andrew Bruce. Robust wavelet denoising. *IEEE Transactions on Signal Processing*, 49(6):1146–1152, 2001.

[46] Viraj Shah and Chinmay Hegde. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *ICASSP*, 2018.

[47] Emil Y Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine & Biology*, 53(17):4777, 2008.

[48] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *ICLR*, 2024.

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[50] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *ICLR*, 2023.

[51] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *NeurIPS*, 34, 2021.

[52] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

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

[54] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023.

[55] 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.

[56] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *CVPR*, pages 25456–25467, June 2024.

[57] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *CVPR*, pages 25456–25467, 2024.

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

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[60] Zixiang Zhao, Haowen Bai, Jiangshe Zhang, Yulun Zhang, Shuang Xu, Zudi Lin, Radu Timofte, and Luc Van Gool. Cddfuse: Correlation-driven dual-branch feature decomposition for multi-modality image fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5906–5916, June 2023.

[61] Zixiang Zhao, Haowen Bai, Yuanzhi Zhu, Jiangshe Zhang, Shuang Xu, Yulun Zhang, Kai Zhang, Deyu Meng, Radu Timofte, and Luc Van Gool. Ddfm: Denoising diffusion model for multi-modality image fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8082–8093, October 2023.

[62] Nicolas Zilberstein, Morteza Mardani, and Santiago Segarra. Repulsive latent score distillation for solving inverse problems. In *ICLR*, 2025.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: All contributions are clearly separated in the abstract and introduction. Ablation studies confirm that each component of our approach contributes meaningfully to the overall performance.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have a limitation section in section 6

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We provide the full proof of our propositions in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We discuss all necessary details to reimplement our method in section 4 and give detailed information on the hyperparameters used for each experiment in subsection A.4.

Guidelines:

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

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

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

Answer: [Yes]

Justification: The datasets we used are openly accessible and we open sourced our code.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide detailed settings for the parameters we used for our method and the baselines in subsection A.4.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We have a statistical relevance section in the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We give detailed information about runtime on the hardware we used in the Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification:

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Our methods has a similar societal impact compared to other image restoration methods, which we highlight in the discussion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our approach is training free and the data and model weights, which have been used are already publicly accessible.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We credit all authors, owners, and creators of the assets we used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: We do not introduce new assets, because our method is training free and the data publicly available.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We did not use crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We did not use crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: We did not use LLMs in any important, original or non-standard component of the core methods.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Supplementary Material

In the following, we provide detailed line-by-line derivations of the mathematical formulations used in the paper, as well as additional implementation details and experimental results.

A Derivations

A.1 Derivation of flow-based variational formulation

The linear conditional flow and it's corresponding velocity are defined as:

$$\psi_t(x_0 \mid \epsilon) = (1-t)\, x_0 + t\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \ , \tag{17}$$

$$u_t(x_t|\epsilon) = \frac{\mathrm{d}\psi_t}{\mathrm{d}t}(\psi_t^{-1}(x_t|\epsilon)|\epsilon) \ . \tag{18}$$

The score of the noisy variational distribution can be analytically computed with:

$$q(x_t|y) = \mathcal{N}((1-t)\mu_x, t^2\, I) \ , \tag{19}$$

$$\nabla_{x_t} \log q(x_t|y) = -\frac{\epsilon}{t} \ . \tag{20}$$

We compute $\frac{\mathrm{d}\psi_t}{\mathrm{d}t}(x_0|\epsilon) = -x_0 + \epsilon$ and $\psi_t^{-1}(x_t|\epsilon)$ and insert it into Equation 18:

$$u_t(x_t|\epsilon) = \frac{\epsilon - x_t}{1-t} \ . \tag{21}$$

Solving Equation 21 for ϵ and inserting in Equation 20 gives:

$$\nabla_{x_t} \log q(x_t|y) = -\frac{(1-t)u_t(x_t|\epsilon) + x_t}{t} \ . \tag{22}$$

For the learned velocity $v_\theta(x_t, t)$ a similar approximation holds:

$$v_\theta(x_t, t) \approx \frac{-t\nabla_x \log p(x_t) - x_t}{1-t} \ . \tag{23}$$

Hence, we can approximate the score of the noisy prior with our learned velocity field v_θ

$$\nabla_{x_t} \log p(x_t) \approx -\frac{(1-t)v_\theta(x_t, t) + x_t}{t} \ , \tag{24}$$

and we see that for $\omega(t) = \frac{t}{1-t}$ we obtain the conditional flow matching objective for $\mathcal{R}(x)$. We therefore set $\omega(t) = \frac{t}{1-t}$ and end up at our final objective:

$$\arg\min_{q(x_0|y)} \underbrace{\mathbb{E}_{q(x_0|y)}\left[\frac{\|y - f(\mu_x)\|^2}{2\nu^2}\right]}_{\mathcal{D}(x,y)} + \underbrace{\int_0^T \ \mathbb{E}_{q(x_t|y)}\left[\|v_\theta(x_t,t) - u_t(x_t|\epsilon)\|^2\right] dt}_{\mathcal{R}(x)} \ . \tag{25}$$

Again, the gradient step for the regularizer becomes:

$$\nabla_{\mu_x}\mathcal{R}(x) = \mathbb{E}_{t, q(x_t|y)}\left[v_\theta(x_t, t) - u_t(x_t|\epsilon)\right] \ . \tag{26}$$

A.2 Derivation of trajectory adjusted flow-based variational formulation

To achieve the proposed trajectory adjustment, we modify the forward process to:

$$\hat{x}_1 = x_{t+dt} + (1-t-dt)v_\theta(x_{t+dt}, t+dt) \ , \tag{27}$$

$$x_t = (1-t)\mu_x + t\,\underbrace{(\alpha\hat{x}_1 + \sqrt{1-\alpha^2}\epsilon)}_{\hat{\epsilon}} \ , \tag{28}$$

where $\hat{x}_1$ is the noise vector prediction from the last optimization iteration. This induces a variational distribution:

$$q(x_t \mid y) = \mathcal{N}\left((1-t)\mu_x + t\alpha\hat{x}_1,\, t^2(1-\alpha^2)I\right) \ , \tag{29}$$

leading to a score of

$$\nabla_{x_t} \log q(x_t \mid y) = -\frac{1}{t^2(1-\alpha^2)} \cdot t\sqrt{1-\alpha^2}\epsilon = -\frac{\epsilon}{t\sqrt{1-\alpha^2}} \; . \tag{30}$$

The velocity field is again computed by Equation 18. We start by defining the flow:

$$\psi_t(x_0 \mid \epsilon) = (1-t)x_0 + t\left(\alpha\hat{x}_1 + \sqrt{1-\alpha^2}\epsilon\right) \; . \tag{31}$$

The resulting derivative reads

$$\frac{d}{dt}\psi_t(x_0 \mid \epsilon) = \alpha\hat{x}_1 - x_0 + \sqrt{1-\alpha^2}\epsilon \; , \tag{32}$$

and the inverse becomes

$$x_0 = \psi_t^{-1}(x_t \mid \epsilon) = \frac{x_t - t\alpha\hat{x}_1 - t\sqrt{1-\alpha^2}\epsilon}{1-t} \; . \tag{33}$$

Plugging these results into Equation 18:

$$u_t(x_t \mid \epsilon) = \frac{\alpha\hat{x}_1 + \sqrt{1-\alpha^2}\epsilon - x_t}{1-t} \; . \tag{34}$$

A.3 Derivation of Score from Flow

The score matching objective reads as:

$$\nabla_{x_t} \ln p_t(x_t) = \arg\min_{\theta} \mathbb{E}_{t\sim\mathcal{U}[0,1],x_0\sim p_0,\epsilon\sim\mathcal{N}(0,I)}\left[w(t)\cdot\left\|s_\theta(x_t,t) + \frac{1}{\sigma(t)^2}\left(x_t - \mu(x_0,t)\right)\right\|^2\right], \tag{35}$$

where,

$$-\frac{1}{\sigma(t)^2}\left(x_t - \mu(x_0,t)\right) = \nabla_{x_t}\log p_t(x_t \mid x_0), \tag{36}$$

with $p_t(x_t \mid x_0) = \mathcal{N}(\mu_t(x_0), \sigma_t^2 I)$. Note that as usual we assume $\mu_t(x_0)$ being linear in x_0. Equation 35 is solved by:

$$\nabla_{x_t}\log p_t(x_t) = \mathbb{E}_{p_t(x_0|x_t)}\left[\nabla_{x_t}\log p_t(x_t|x_0)\right], \tag{37}$$

and can be written as:

$$\nabla_{x_t}\log p_t(x_t) = \frac{-(x_t - \mu(\mathbb{E}[x_0 \mid x_t], t))}{\sigma(t)^2}. \tag{38}$$

In the case of OT flow-matching, we obtain

$$x_t = (1-t)x_0 + tx_1, \tag{39}$$

$x_1 \sim \mathcal{N}(0, \mathrm{Id})$ and $p(x_t|x_0) = \mathcal{N}((1-t)x_0, t^2)$. The optimal velocity under the flow matching loss is given by:

$$v^*(x_t, t) = \mathbb{E}[x_1 - x_0 \mid x_t]. \tag{40}$$

Expressing $x_1 = \frac{x_t-(1-t)x_0}{t}$, we can insert into Equation 40 and obtain:

$$\mathbb{E}[x_0 \mid x_t] = x_t - t\mathbb{E}[x_1 - x_0 \mid x_t]. \tag{41}$$

Inserting in Equation 38 leads to:

$$\nabla_{x_t}\log p_t(x_t) = -\frac{x_t - (1-t)(x_t - t\mathbb{E}[x_1 - x_0 \mid x_t])}{t^2}, \tag{42}$$

which for $v^*(x_t,t) = \mathbb{E}[x_1 - x_0 \mid x_t]$ reads as:

$$\nabla_{x_t}\log p(x_t) \approx -\frac{(1-t)v_\theta(x_t,t) + x_t}{t}. \tag{43}$$

A.4 Implementation details

Flow Model and Regularizer Settings. As flow matching model, we us Stable Diffusion 3.5-Medium, which has been released under the Stability Community License. The classifier-free guidance scale is set to 2 for all experiments. To minimize the regularization term, we use stochastic gradient descent with a learning rate of 1.

Data Likelihood Term. We use stochastic gradient descent for the minimization of the data term towards hard data consistency. For numerical stability, the squared error is summed over all measurements instead of computing the mean. The learning rate has to be adjusted accordingly, to compensate for the varying number of measurements y. Moreover, the minimization is terminated with early stopping once the likelihood term reaches $1 \times 10^{-4} \cdot \text{len}(y)$, to not overfit the noise in the image observation.

Super-resolution. We employ bicubic downsampling as the forward operator, as implemented in [53]. The learning rate is set to 12 for ×12 super-resolution and to 6 for ×8 super-resolution.

Motion Deblurring. A different motion blur kernel is created for each sample using the *MotionBlur* package [8], available via github, with kernel size 61 and intensity 0.5. The learning rate for our data term optimizer is set to 10^{-1}.

Inpainting. For inpainting on FFHQ we always use the same rectangular mask at a fixed position, chosen such that it roughly masks out the right side of the face (Figure 3). For DIV2k we also use a fixed mask for all samples, consisting of six randomly generated rectangles (Figure 6).

Data. We use the publicly available Flickr Faces High Quality dataset [24], which is realeased under the Creative Commons BY 2.0 License and the DIV2K dataset [3], which is released under a research only license. For FFHQ we use the first 1000 samples of the evaluation dataset and for DIV2K we use the 800 training samples. We downscale both datasets to 768×768 px by applying bicubic sampling so that the shorter edge of the frame has 768 px and apply central cropping afterwards.

A.5 Baselines

For comparability, all baselines use Stable Diffusion 3.5-Medium and the same task definitions as in A.4.

FlowDPS [26] The standard FlowDPS implementation [27] is applied with 50 NFE, a classifier-free guidance scale of 2, and step sizes of 15 for inpainting and 10 for all other tasks.

FlowChef [39] Additionally, [27] is employed for FlowChef as well, using 200 NFE for inpainting and 50 NFE for all other tasks, a classifier-free guidance scale of 2, and a step size of 1 for all tasks.

Repulsive Score Distillation (RSD) [62]. We implement RSD for flow-matching models by applying Proposition 1 with $\omega(t) = t$, resulting in a weighting term consistent with the original RSD approach. However, we omit the pixel-space augmentation as it negatively affected performance when combined with the SD3 VAE. Consistent with the original findings from RSD, we observed that incorporating the repulsive term improves sample diversity but reduces fidelity. Therefore, we set the repulsive term to 0 for all results presented in the table, employing it exclusively for comparing posterior variances.

ReSample [48] We re implement ReSample for flow-matching by setting $\bar{\alpha}_t = \frac{(1-t)^2}{t^2+(1-t)^2}$. Furthermore, we compute the hard-data consistency at every iteration as larger skip steps seem to harm performance. We set the learning rate of the data term optimizer to 15 for all inverse problems.

PSLD [42] Our attempt to adapt PSLD following [27]—using 500 NFE, a classifier-free guidance scale of 2, and step sizes of 1 (×12 super-resolution), 0.5 (×8 super-resolution and motion deblurring) and 0.1 (inpainting)—did not yield meaningful results.

A.6 Regularizer weighting

Figure 1 displays the mean and standard deviation of the conditional flow matching loss $\mathcal{L}_{CFM}$ as a function of t, estimated over 100 samples. The loss function starts with high values at $t = 1$, decreases over time, but then starts to rise again, and when reaching $t \approx 0.2$ even exceeds its initial value . The rising loss when approaching $t = 0$ is due, in part, to the increasing difficulty of distinguishing high-frequency image content from residual noise. Another factor is that near $t = 0$ the model operates in a highly sensitive regime where small prediction errors can cause disproportionately large deviations from the target, making accurate flow estimation particularly challenging in the final stages of the trajectory. We therefore modulate the regularization term according to the model error. Different weighting functions for $f(\mathcal{L}_{CFM})$ could be chosen that fulfill the condition $\lambda_{\mathcal{R}(t=0)} = 0$. We simply take the reciprocal of the model error $\lambda_{\mathcal{R}(t)} = \mathcal{L}^{-1}_{CFM,t}$ as the regularization weight while $t \geq 0.2$, then set it to 0 for $t < 0.2$. An alternative would be to shift the reciprocal of $\mathcal{L}^{-1}_{CFM,t}$ by $\mathcal{L}^{-1}_{CFM,t=0}$, such that $\lambda_{\mathcal{R}(t)} = \mathcal{L}^{-1}_{CFM,t} - \mathcal{L}^{-1}_{CFM,t=0}$. In Table 1 we compare our default weighting with this variant, denoted as λ_{shift}.

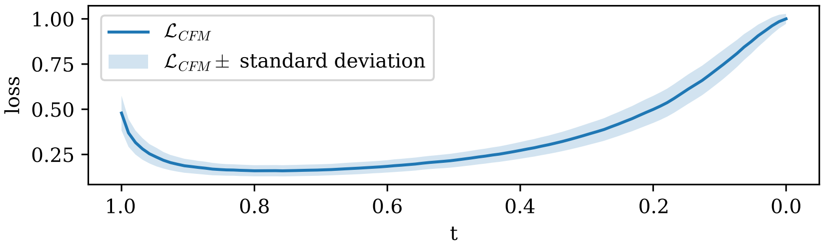


Figure 1: The Flow-Matching loss over time t.

Table 1: **Quantitative results** with 50 NFE and $\sigma_\nu = 0.01$. We compare different weighting functions $\lambda_{\mathcal{R}(t)}$ based on the model error

	SR ×8				SR ×12				Motion Deblurring				Inpainting			
Method	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑
FFHQ 768×768																
λ_{shift}	0.246	27.4	**0.793**	**29.91**	0.286	24.4	**0.766**	**28.19**	0.237	14.5	**0.790**	**29.84**	**0.180**	**8.2**	**0.828**	23.58
Ours	**0.213**	**13.3**	0.777	29.54	**0.271**	**16.2**	0.740	27.71	**0.236**	**10.7**	0.772	29.61	0.184	8.7	0.828	**23.69**
DIV2K 768×768																
λ_{shift}	0.379	30.4	**0.625**	**23.58**	0.434	37.5	0.522	**21.40**	0.337	25.8	**0.664**	**24.54**	**0.151**	**9.0**	**0.819**	**23.79**
Ours	**0.353**	**26.5**	0.607	23.30	**0.421**	**32.1**	**0.525**	21.39	**0.315**	**21.1**	0.653	24.44	0.163	11.0	0.815	23.75

A.7 Effect of captioning

Given the diversity of DIV2k, we use DAPE [57] to generate captions for it and include them in the prompt *A high quality photo of [DAPE caption].* For FFHQ we always prompt with *A high quality photo of a face.*. The effect of the text prompt is to increase the likelihood of our sample under the prior of the (pre-trained, frozen) image generator. For comparison, we also ran experiments without data specific captions, where we always used the generic prompt *A high quality photo.* Results are shown in Table 2

Table 2: Quantitative results with 50 NFE and $\sigma_\nu = 0.01$. We compare our version with data-specific captions and a version without captions.

	FFHQ				DIV2K			
Method	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑
wo captions	0.278	17.0	0.734	27.66	0.488	51.5	**0.546**	**21.82**
Ours	**0.271**	**16.2**	**0.740**	**27.71**	**0.421**	**32.1**	0.525	21.39

A.8 Additional Experimental Results

We present the experimental results from the main paper in Table 3, now augmented with sample-wise standard deviations for all metrics except FID.

Table 3: **Quantitative results** with 50 NFE and $\sigma_\nu = 0.01$ – Super-resolution (×8 and ×12).

	SR ×8				SR ×12			
Method	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑
				FFHQ 768×768				
ReSample	0.400 ± 0.069	55.6	**0.815 ± 0.051**	26.37 ± 1.00	0.474 ± 0.078	80.3	**0.786 ± 0.056**	25.47 ± 1.16
FlowDPS	0.374 ± 0.107	38.5	0.756 ± 0.075	29.24 ± 2.04	0.413 ± 0.107	44.0	0.741 ± 0.074	28.05 ± 2.06
RSD	0.391 ± 0.079	51.7	0.776 ± 0.052	**29.69 ± 2.04**	0.462 ± 0.093	71.7	0.743 ± 0.059	**28.11 ± 2.00**
FlowChef	0.341 ± 0.083	30.5	0.760 ± 0.064	28.42 ± 2.22	0.373 ± 0.084	46.5	0.730 ± 0.068	27.00 ± 2.07
Ours	**0.213 ± 0.056**	**13.3**	0.777 ± 0.051	29.54 ± 2.02	**0.271 ± 0.071**	**16.2**	0.740 ± 0.058	27.71 ± 2.00
				DIV2K 768×768				
ReSample	0.533 ± 0.130	55.0	**0.625 ± 0.132**	22.34 ± 2.27	0.643 ± 0.152	88.1	**0.562 ± 0.151**	20.85 ± 3.02
FlowDPS	0.476 ± 0.129	44.4	0.567 ± 0.139	23.01 ± 3.01	0.547 ± 0.139	54.0	0.528 ± 0.146	21.79 ± 2.94
RSD	0.539 ± 0.121	60.9	0.591 ± 0.124	**23.45 ± 2.96**	**0.684 ± 0.137**	95.7	0.523 ± 0.132	**21.96 ± 2.86**
FlowChef	0.490 ± 0.116	36.5	0.539 ± 0.137	21.84 ± 2.96	0.525 ± 0.118	43.8	0.492 ± 0.145	20.52 ± 2.85
Ours	**0.353 ± 0.112**	**26.5**	0.607 ± 0.127	23.30 ± 2.90	**0.421 ± 0.131**	**32.1**	0.525 ± 0.136	21.39 ± 2.67

Table 4: **Quantitative results** with 50 NFE and $\sigma_\nu = 0.01$ – Motion deblurring and in-painting.

	Motion Deblurring				In-painting			
Method	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑
				FFHQ 768×768				
ReSample	0.457 ± 0.087	82.9	**0.788 ± 0.058**	25.45 ± 1.46	0.366 ± 0.053	70.8	0.827 ± 0.033	21.83 ± 1.68
FlowDPS	0.431 ± 0.117	54.3	0.732 ± 0.078	27.64 ± 2.20	0.344 ± 0.060	42.5	0.771 ± 0.048	19.19 ± 3.19
RSD	0.458 ± 0.098	77.3	0.743 ± 0.059	27.67 ± 2.47	0.478 ± 0.082	73.3	0.736 ± 0.048	21.97 ± 2.58
FlowChef	0.406 ± 0.093	40.2	0.716 ± 0.072	25.81 ± 2.61	0.394 ± 0.069	69.8	0.780 ± 0.051	18.18 ± 2.84
Ours	**0.236 ± 0.070**	**10.7**	0.772 ± 0.055	**29.61 ± 2.24**	**0.184 ± 0.038**	**8.7**	**0.828 ± 0.029**	**23.69 ± 2.77**
				DIV2K 768×768				
ReSample	0.556 ± 0.146	79.7	0.617 ± 0.134	21.79 ± 2.52	0.285 ± 0.073	51.9	0.796 ± 0.067	22.68 ± 1.84
FlowDPS	0.558 ± 0.153	65.5	0.536 ± 0.148	21.88 ± 3.02	0.328 ± 0.103	29.2	0.692 ± 0.112	21.71 ± 2.67
RSD	0.638 ± 0.156	97.6	0.551 ± 0.136	22.10 ± 3.07	0.464 ± 0.112	63.9	0.678 ± 0.077	23.23 ± 2.21
FlowChef	0.561 ± 0.123	49.6	0.486 ± 0.148	19.90 ± 3.06	0.489 ± 0.148	58.3	0.659 ± 0.131	20.87 ± 2.65
Ours	**0.315 ± 0.107**	**21.1**	**0.653 ± 0.121**	**24.44 ± 3.05**	**0.163 ± 0.053**	**11.0**	**0.815 ± 0.054**	**23.75 ± 2.74**

A.9 FLAIR in Pixel Space

Table 5: **Quantitative results** with 50 NFE and $\sigma_\nu = 0.5\%$ – In-painting and Super-resolution (×8).

	Inpainting				SR ×8			
Method	LPIPS↓	FID↓	SSIM↑	PSNR↑	LPIPS↓	FID↓	SSIM↑	PSNR↑
DDNM	0.158 ± 0.042	26.9	0.732 ± 0.037	18.31 ± 2.94	0.199 ± 0.052	31.9	0.635 ± 0.079	23.59 ± 1.64
DPS	0.195 ± 0.064	30.2	0.689 ± 0.077	20.49 ± 2.81	0.172 ± 0.058	27.8	0.658 ± 0.088	24.59 ± 2.04
MM	0.161 ± 0.054	28.8	0.728 ± 0.062	20.59 ± 3.37	0.172 ± 0.051	29.1	0.669 ± 0.083	24.65 ± 1.97
ΠGDM	0.195 ± 0.064	30.2	0.689 ± 0.077	20.49 ± 2.81	0.157 ± 0.052	26.5	0.677 ± 0.084	24.98 ± 2.07
FLAIR	**0.097 ± 0.035**	**14.2**	**0.831 ± 0.031**	**21.87 ± 2.66**	**0.143 ± 0.039**	**22.9**	**0.712 ± 0.076**	**25.93 ± 1.96**

We additionally implement our method including DTA and $\lambda_{\mathcal{R}(t)} = \mathcal{L}_{CFM,t}^{-1}$ (0 for $t < 0.2$) in pixel space using the flow model from [33], trained on CelebA-HQ resized (256x256). For comparison, we rephrase score based baselines to flow following [26] and evaluate all on 1000 samples from the dataset on super-resolution and inpainting. The methods are hyperparameter-tuned to DDNM [53] (likelihood weight 4 for inpainting | 1 for SR8), DPS [11] (64 | 512), Moment Matching [43] (4 | 8), ΠGDM [50] (64 | 8) and pixel space FLAIR (0.5 | 32 and regularizer weight 0.4). As shown in Table 5, our method outperforms previous works also in pixel space, demonstrating its broader applicability.

A.10 Runtime Analysis

We compare the runtime and memory consumption of our method to the baselines. As our hard data consistency can strongly influence the runtime, we also provide measurements with the number of data term steps ≤ 5 and additionally a fast version using a "tinyVAE" of SD3. To validate that the

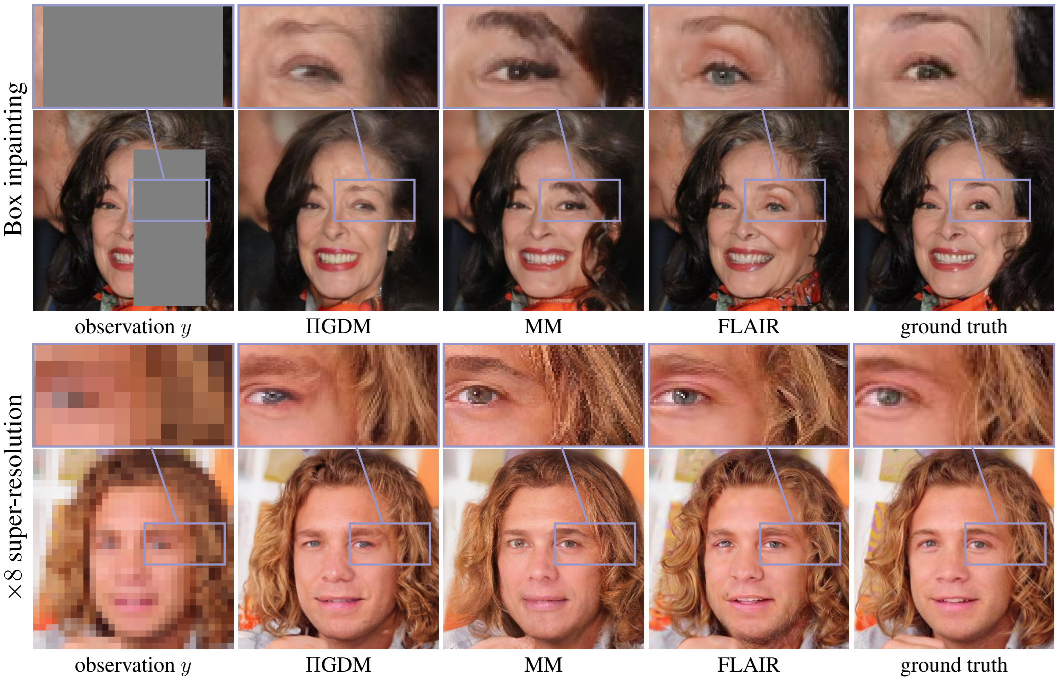


Figure 2: **Qualitative comparison.** FLAIR in pixel space produces posterior samples.

usage of the tinyVAE or less steps does not degrades the performance noticeably we also provide a metrics for x12 Super Resolution on 100 samples of FFHQ:

Table 6: Comparison of different methods in terms of runtime and memory usage. We validate the use of less data steps and a "tinyVAE" on ×12 super resolution on 100 samples of the FFHQ dataset.

Method	**Runtime (s)** $\downarrow$	**Memory (MB)** $\downarrow$	**LPIPS** $\downarrow$	**PSNR** $\uparrow$
Resample	88.02	19009.2	0.461	25.31
FlowDPS	34.15	12228.6	0.404	27.74
RSD (no repulsion)	21.19	12400.0	0.462	28.11
FlowChef	15.23	12227.72	0.361	26.56
FLAIR (HDC, large VAE)	172.34	12389.4	0.259	27.42
FLAIR (HDC, tiny VAE)	40.77	5960.2	0.256	27.59
FLAIR (5 data term steps, tiny VAE)	22.46	5960.2	0.264	27.61

A.11 Ensembling Experiment

To highlight that our model can also be used to obtain good MMSE estimates, we also conducted ensemble predictions by running posterior sampling 8 times and averaging the result. The results show that ensembling increases PSNR values, but reduces LPIPS and confirms that our samples are indeed distributed around the posterior mean and that results very close to the posterior mean like the baseline methods are perceptually further away (LPIPS) from the ground truth compared to our samples.

A.12 Statistical Relevance

Our method is training-free, and the variance in reconstructed images is **intentional**, reflecting the stochasticity of our sampling process rather than instability. All methods are evaluated with identical random seeds to ensure fair comparison. We compute metrics over 1000 samples for FFHQ and 800 samples for DIV2K. Perceptual FID (pFID) is evaluated on 256 × 256 patches, resulting in 9000 and 7200 samples, respectively. Table 3 in Appendix A.8 reports means and standard deviations over multiple samples.

Table 7: **Quantitative results** – Super-resolution (×8 and ×12). We report PSNR↑ and LPIPS↓. For ensembling we averaged 8 independent predictions of the corresponding methods. It can be seen that PSNR improves for all methods when ensembling. However, FLAIR shows the biggest gain, which means that our samples are indeed distributed around the mean and feature a higher variance compared to the baselines.

	SR ×8		SR ×12	
Method	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
	DIV2K			
FlowDPS	22.53	0.4837	21.47	0.5524
FlowDPS (8x ensemble)	23.28	0.5157	22.06	0.5995
FlowChef	21.44	0.4898	20.20	0.5228
FlowChef (8x ensemble)	22.60	0.5502	21.60	0.5931
FLAIR	22.83	**0.3627**	21.05	**0.4270**
FLAIR (8x ensemble)	**23.79**	0.4244	**22.27**	0.4930
	FFHQ			
FlowDPS	29.02	0.3659	27.74	0.4036
FlowDPS (8x ensemble)	30.12	0.3267	28.65	0.3749
FlowChef	28.05	0.3303	26.56	0.3609
FlowChef (8x ensemble)	29.54	0.3267	28.15	0.3602
FLAIR	29.36	**0.2028**	27.42	**0.2594**
FLAIR (8x ensemble)	**30.94**	0.2457	**29.00**	0.2999

We also evaluated 100 FFHQ samples and 80 DIV2K samples, sampling three reconstructions per input for each method. We report the mean of each metric across all samples and the standard deviation of the means.

Table 8: Statistical evaluation on **FFHQ** for ×**8 Super Resolution**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS ↓	FID ↓	SSIM ↑	PSNR ↑
FlowDPS	0.370 ± 0.0012	70.7 ± 1.45	0.755 ± 0.0010	28.98 ± 0.008
RSD	0.4678 ± 0.0001	102.9 ± 0.05	0.7362 ± 0.0001	28.45 ± 0.001
FlowChef	0.3316 ± 0.0055	63.5 ± 1.45	0.7593 ± 0.0027	28.12 ± 0.072
Ours	**0.2039** ± 0.0048	**40.5** ± 0.94	**0.7970** ± 0.0228	**29.74** ± 0.668

Table 9: Statistical evaluation on **FFHQ** for ×**12 Super Resolution**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS ↓	FID ↓	SSIM ↑	PSNR ↑
FlowDPS	0.4073 ± 0.0002	77.6 ± 1.05	0.7391 ± 0.0006	27.71 ± 0.016
RSD	0.5039 ± 0.0002	119.0 ± 0.12	0.7217 ± 0.0001	27.08 ± 0.001
FlowChef	0.3626 ± 0.0050	81.1 ± 1.03	0.7283 ± 0.0027	26.62 ± 0.059
Ours	**0.2593** ± 0.0023	**45.8** ± 1.51	**0.7582** ± 0.0252	**27.81** ± 0.660

A.12.1 t-Test Analysis

We further performed paired **t-tests** on the LPIPS scores between FlowDPS and FLAIR. The null hypothesis states that the mean LPIPS scores are the same for both methods. In all settings, we reject the null hypothesis ($p < 0.001$), confirming the statistical significance of our improvements see Table 16.

Table 10: Statistical evaluation on **FFHQ** for **Motion Blur**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS $\downarrow$	FID $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
FlowDPS	0.4140 ± 0.0030	83.83 ± 1.00	0.7383 ± 0.0006	27.47 ± 0.05
RSD	0.4515 ± 0.0001	108.75 ± 0.08	0.7437 ± 0.0001	27.40 ± 0.00
FlowChef	0.4019 ± 0.0007	74.89 ± 0.69	0.7178 ± 0.0022	25.50 ± 0.04
Ours	**0.2196** ± 0.0080	**38.8** ± 2.25	**0.7964** ± 0.0319	**30.10** ± 0.96

Table 11: Statistical evaluation on **FFHQ** for **Inpainting**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS $\downarrow$	FID $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
FlowDPS	0.3315 ± 0.0015	74.00 ± 0.58	0.7755 ± 0.0007	19.06 ± 0.11
RSD	0.4601 ± 0.0003	103.02 ± 0.03	0.7430 ± 0.0000	22.19 ± 0.01
FlowChef	0.3771 ± 0.0013	102.22 ± 0.26	0.7888 ± 0.0016	18.42 ± 0.25
Ours	**0.1761** ± 0.0012	**33.23** ± 2.02	**0.8423** ± 0.0172	**24.07** ± 0.80

A.13 Additional Qualitative Examples

To illustrate the visual differences behind the error metrics, we present additional qualitative results for both FFHQ and DIV2k, comparing FLAIR with existing approaches. These examples complement the images in the main paper and highlight the visual fidelity, consistency, and robustness of our method across diverse scenes and different degradations. Figure 9 features a full sized version of the variance figure in section subsection 5.2.

A.14 Failure cases

We observe two main failure modes for FLAIR, see Figure 10. First, we find that super-resolution on DIV2k occasionally results in grainy textures, usually in regions with abundant high-frequency detail and complicated light transport. Potentially, this happens for images which do not have high probability under the prior. We do not observe those artifacts for the FFHQ dataset. Second, we observe a few instances where the strong generative prior hallucinates semantically inconsistent or misaligned structures – especially facial features.

Table 12: Statistical evaluation on **DIV2K** for ×8 **Super Resolution**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS ↓	FID ↓	SSIM ↑	PSNR ↑
FlowDPS	0.5517 ± 0.0046	138.24 ± 1.67	0.5207 ± 0.0022	22.41 ± 0.02
RSD	0.7163 ± 0.0002	181.83 ± 0.15	0.4892 ± 0.0001	21.99 ± 0.00
FlowChef	0.5726 ± 0.0046	145.77 ± 3.31	0.4998 ± 0.0014	21.21 ± 0.04
Ours	**0.3716** ± 0.0161	**88.08** ± 1.43	**0.5991** ± 0.0192	**23.06** ± 0.41

Table 13: Statistical evaluation on **DIV2K** for ×12 **Super Resolution**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS ↓	FID ↓	SSIM ↑	PSNR ↑
FlowDPS	0.6264 ± 0.0045	154.31 ± 0.43	0.4866 ± 0.0024	21.37 ± 0.02
RSD	0.7714 ± 0.0002	198.85 ± 0.12	0.4683 ± 0.0001	21.15 ± 0.00
FlowChef	0.6020 ± 0.0060	151.32 ± 2.45	0.4586 ± 0.0020	20.10 ± 0.03
Ours	**0.4316** ± 0.0151	**101.12** ± 4.51	**0.5236** ± 0.0229	**21.35** ± 0.51

Table 14: Statistical evaluation on **DIV2K** for **Motion Deblur**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS ↓	FID ↓	SSIM ↑	PSNR ↑
FlowDPS	0.6242 ± 0.0082	161.57 ± 3.90	0.4978 ± 0.0028	21.41 ± 0.04
RSD	0.8067 ± 0.0002	216.37 ± 0.36	0.4364 ± 0.0001	20.82 ± 0.00
FlowChef	0.6292 ± 0.0007	158.08 ± 3.03	0.4557 ± 0.0025	19.62 ± 0.05
Ours	**0.3069** ± 0.0036	**77.77** ± 1.89	**0.6596** ± 0.0316	**24.46** ± 0.71

Table 15: Statistical evaluation on **DIV2K** for **Inpainting**. We report mean ± standard deviation over 3 reconstructions per input.

Method	LPIPS ↓	FID ↓	SSIM ↑	PSNR ↑
FlowDPS	0.3738 ± 0.0032	106.58 ± 0.89	0.6579 ± 0.0009	21.06 ± 0.05
RSD	0.4667 ± 0.0003	136.82 ± 0.21	0.6650 ± 0.0001	23.07 ± 0.00
FlowChef	0.5111 ± 0.0019	128.97 ± 0.60	0.6355 ± 0.0006	20.51 ± 0.02
Ours	**0.1729** ± 0.0014	**51.41** ± 0.48	**0.8122** ± 0.0124	**24.06** ± 0.87

Table 16: Paired t-test p-values for LPIPS (FlowDPS vs. Ours). All comparisons are statistically significant.

Dataset	**Task**	**p-value**
DIV2K	SR×8	2.92×10^{-4}
	SR×12	7.19×10^{-4}
	Motion Deblur	5.30×10^{-4}
	Inpainting	1.21×10^{-4}
FFHQ	SR×8	7.05×10^{-5}
	SR×12	6.56×10^{-5}
	Motion Deblur	4.29×10^{-5}
	Inpainting	3.00×10^{-5}

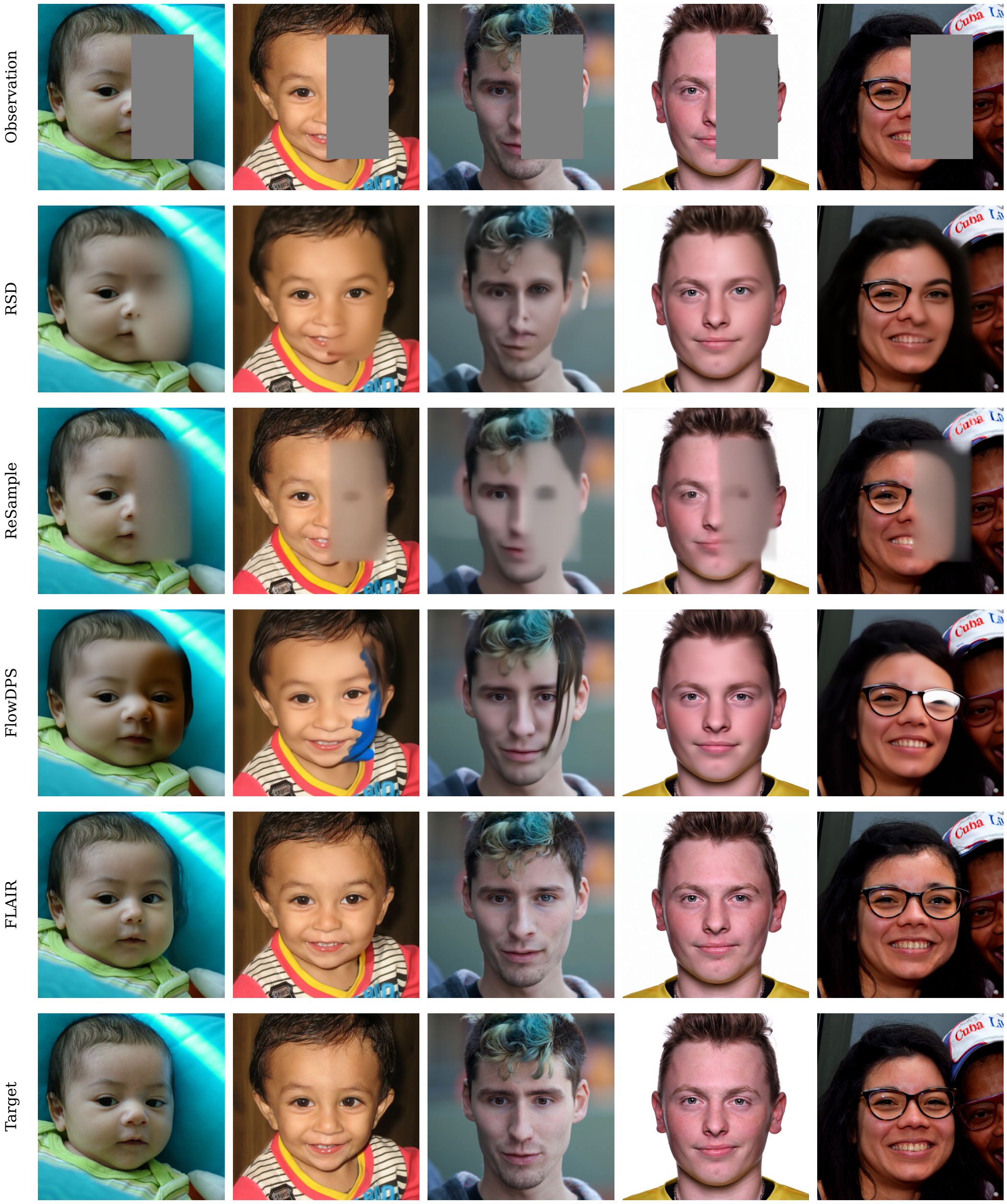


Figure 3: Inpainting results on FFHQ. Shown are observation, reference methods, FLAIR and ground truth. FLAIR produces realistic, high-frequency details while previous works either fail to inpaint the region correctly or collapse to overly smooth solutions.

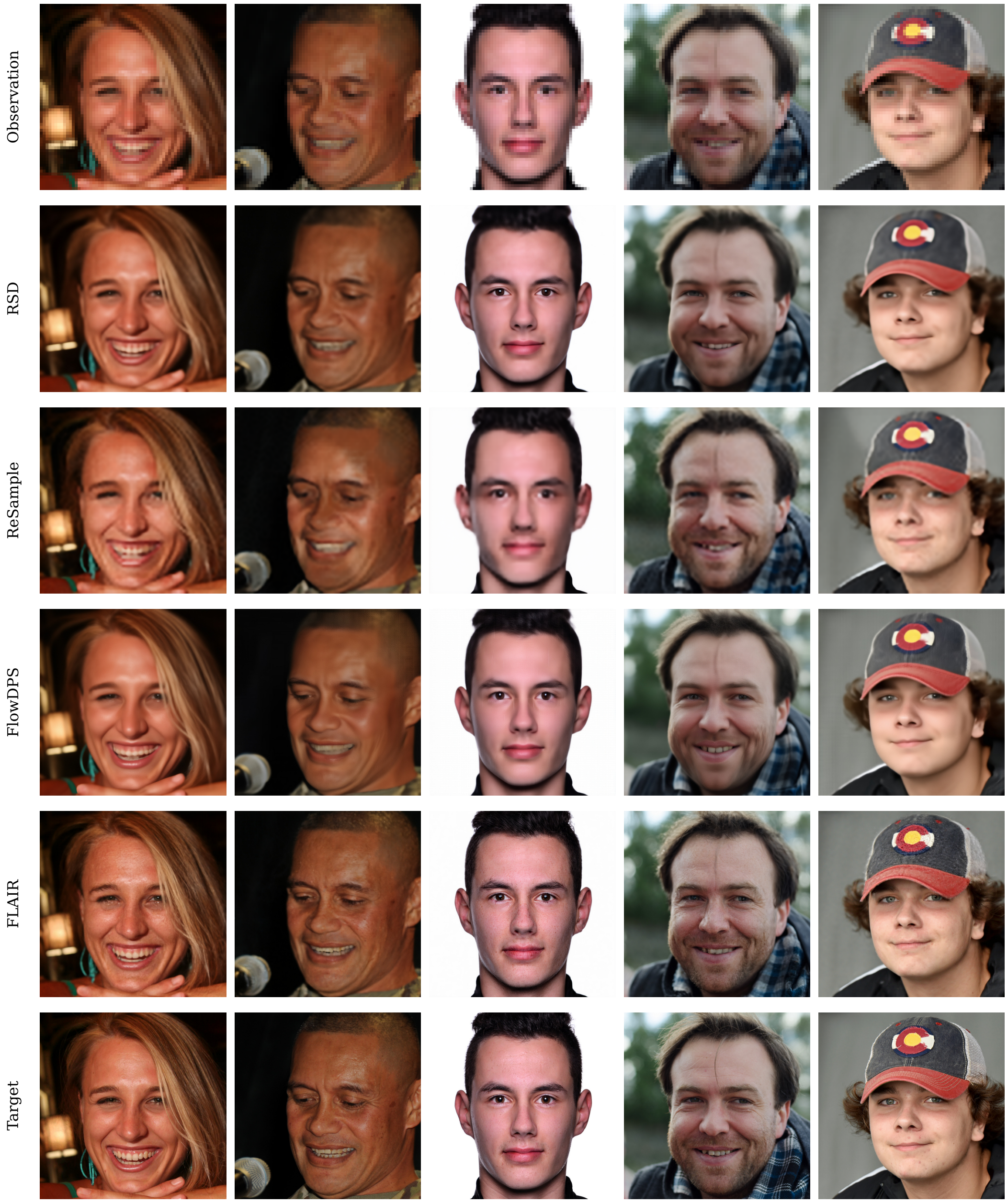


Figure 4: ×12 super-resolution results on FFHQ. Shown are observation, reference methods, FLAIR and ground truth. FLAIR produces sharp and results which still fulfill the data term, whereas the baselines tend to predict blurry images.

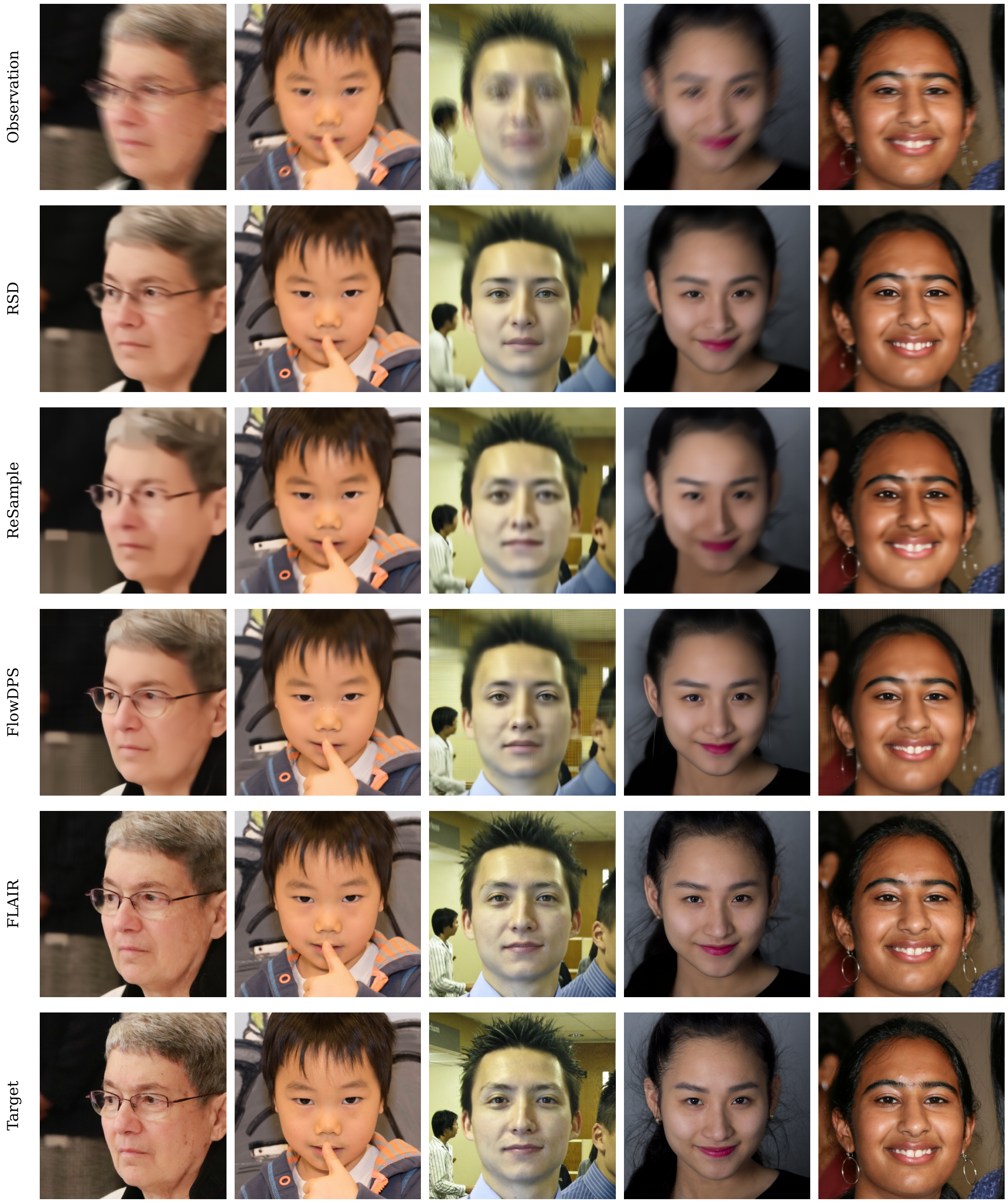


Figure 5: Motion de-blur results on FFHQ. Shown are observation, reference methods, FLAIR and ground truth. FLAIR produces sharp and results which still fulfill the data term, whereas the baselines tend to predict blurry images.

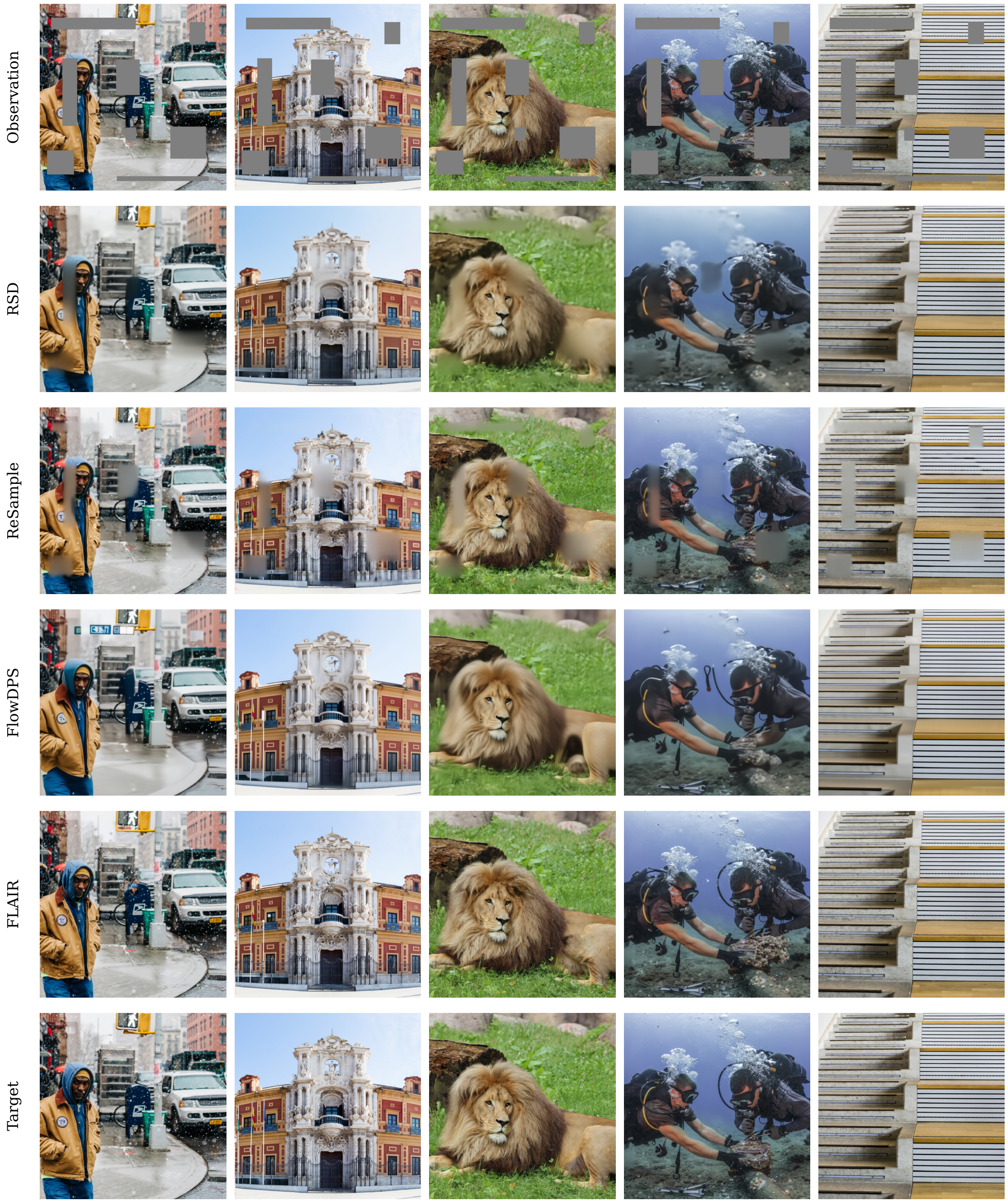


Figure 6: Inpainting results on DIV2k. Shown are observation, reference methods, FLAIR and ground truth. FLAIR produces realistic, high-frequency details while previous works either fail to inpaint the region correctly or collapse to overly smooth solutions. Moreover they do not fit the data term (not inpainted region) very well.

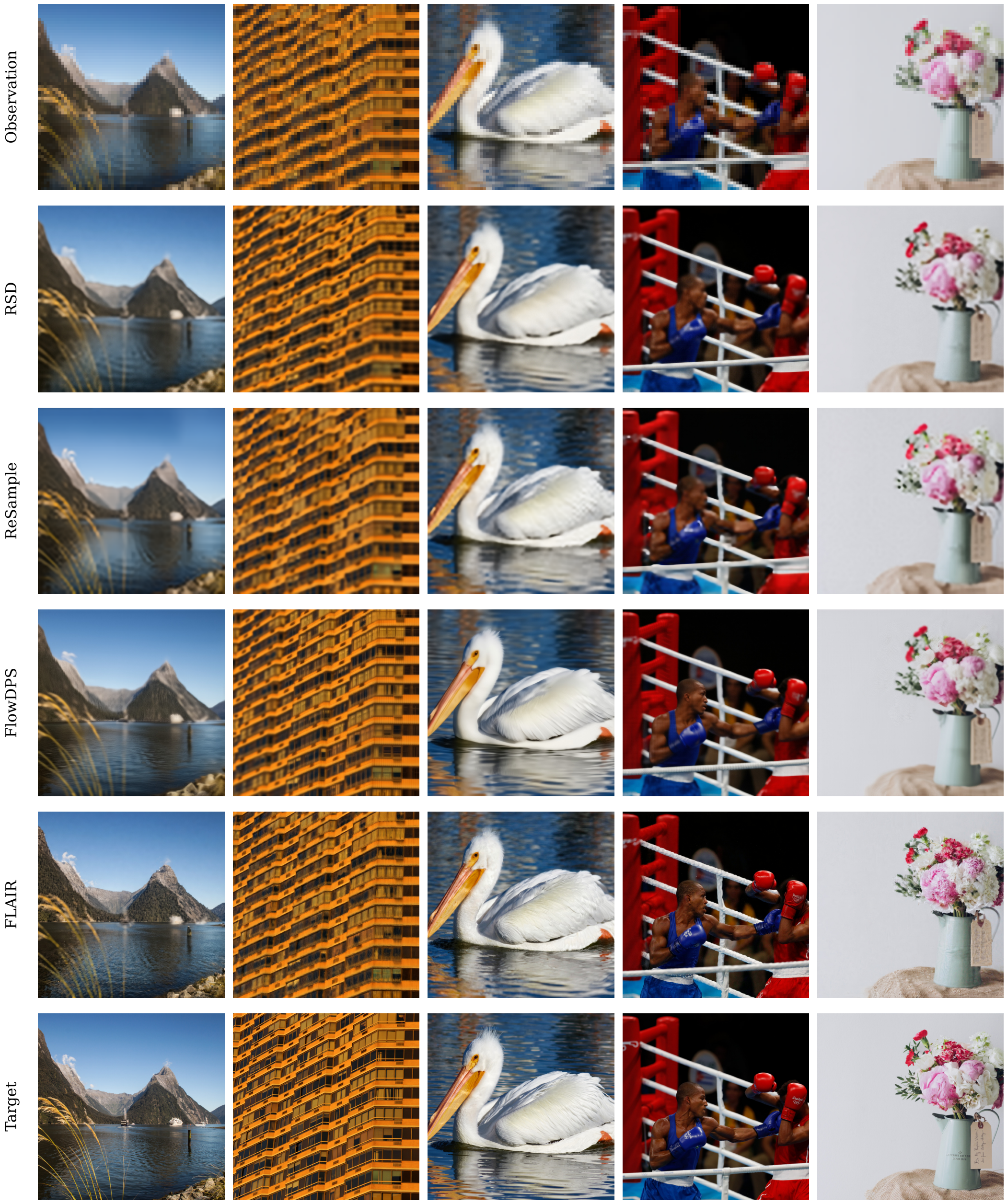


Figure 7: ×12 super-resolution results on DIV2k. Shown are observation, reference methods, FLAIR and ground truth. FLAIR produces sharp and results which still fulfill the data term, whereas the baselines tend to predict blurry images.

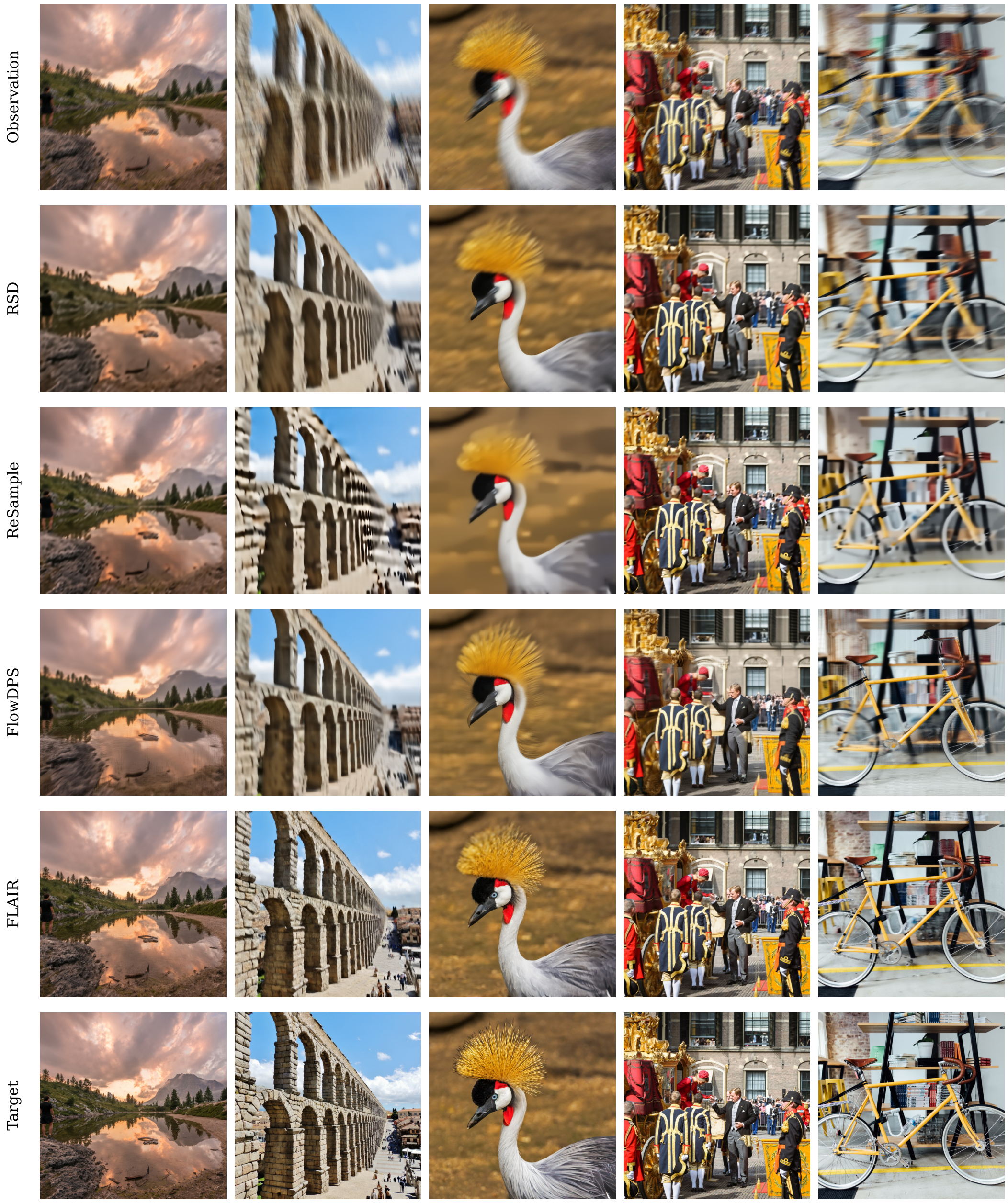


Figure 8: Motion de-blur results results on DIV2k. Shown are observation, reference methods, FLAIR and ground truth. FLAIR produces sharp and results which still fulfill the data term, whereas the baselines tend to predict blurry images.

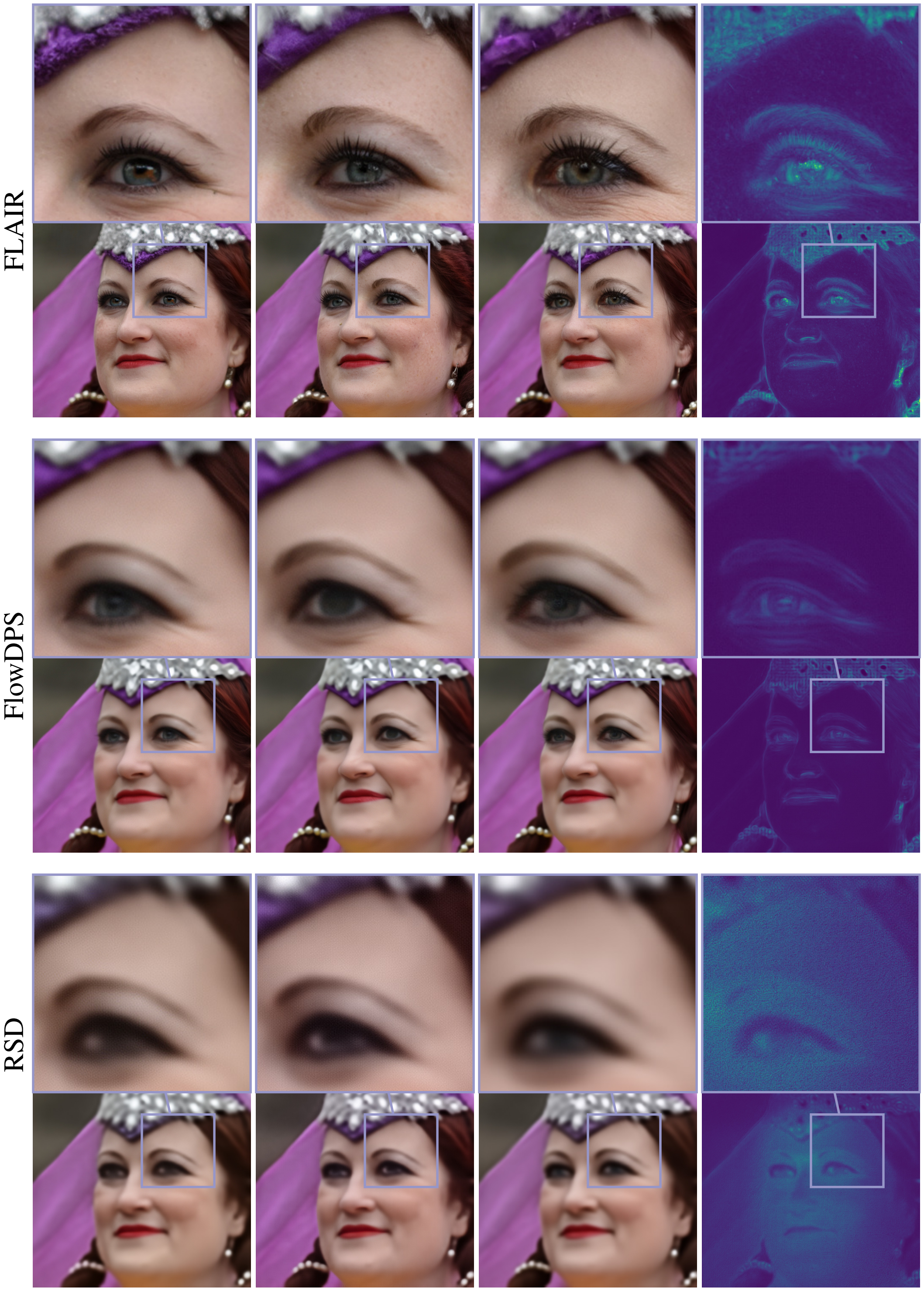


Figure 9: Individual samples for x12 Super Resolution with zoom and std. FLAIR produces varied samples from the posterior. For superresoltion The variance is expected to be mostly in the high frequencies, because the data term limits low frequency variations. The baselines tend to predict very similar looking images with less detail.

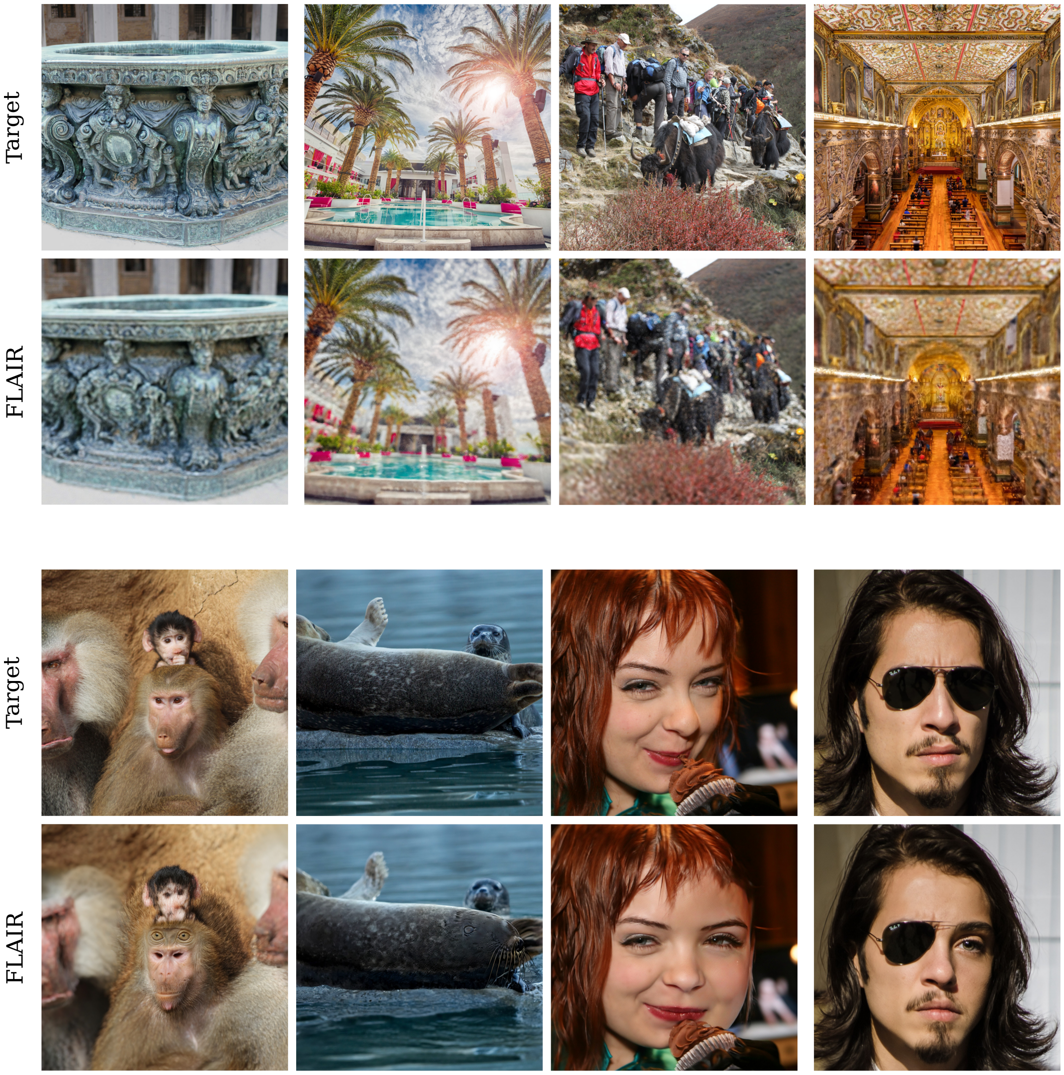


Figure 10: Qualitative failure cases of FLAIR on DIV2k and FFHQ. Top row: grainy results from systematic error. Those errors potentially stem from a weak prior for those images. For example we do not observe them for the FFHQ dataset Bottom row: Semantically inconsistent failures. Sometimes the model lacks the ability to incorporate globally consistent semantics into its restorations.