Consistency Posterior Sampling for Diverse Image Synthesis

Supplementary Material

A. Proofs

Lemma A.1. *The equilibrium distribution of SDE (8) is $\tilde{p}_{1,y}$.*

Proof of Lemma A.1. Under generic conditions, the Langevin dynamics

$$dX_t = -\nabla U(X_t)dt + \sqrt{2}dW_t$$

have the equilibrium $\rho_\infty \propto e^{-U}$. For $\tilde{p}_{1,y}$ in (7) to be the equilibrium, it suffices to verify that

$$\nabla \log \tilde{p}_{1,y} = -(x_1 + \nabla_{x_1} L_y(\Phi(x_1))).$$

This follows by that $\log \gamma(x_1) = -\|x_1\|^2/2 + c$ and $\log p(y|\Phi(x_1)) = -L_y(\Phi(x_1))$. □

Proof of Theorem 4.1. By (4) and (6), we have

$$p_{0,y}(x_0) = \frac{1}{Z_y} p(y|x_0) p_{\text{data}}(x_0), \quad \tilde{p}_{0,y}(x_0) = \frac{1}{\tilde{Z}_y} p(y|x_0) \Phi_\# \gamma(x_0),$$

where

$$Z_y := \int p(y|x_0) p_{\text{data}}(x_0) dx_0, \quad \tilde{Z}_y := \int p(y|x_0) \Phi_\# \gamma(x_0) dx_0.$$

Then, we have

$$\begin{aligned} 2\,\mathrm{TV}(p_{0,y}, \tilde{p}_{0,y}) &= \int |p_{0,y}(x_0) - \tilde{p}_{0,y}(x_0)| dx_0 \\ &\le \int \frac{1}{Z_y} p(y|x_0) \left| p_{\text{data}}(x_0) - \Phi_\# \gamma(x_0) \right| dx_0 + \left| \frac{\tilde{Z}_y - Z_y}{Z_y} \right|. \end{aligned} \tag{A.1}$$

By definition of κ_y in (10), we have $\frac{1}{Z_y} p(y|x_0) \le \kappa_y, \forall x_0$, and thus

$$\int \frac{1}{Z_y} p(y|x_0) \left| p_{\text{data}}(x_0) - \Phi_\# \gamma(x_0) \right| dx_0 \le \kappa_y \int \left| p_{\text{data}}(x_0) - \Phi_\# \gamma(x_0) \right| dx_0.$$

Meanwhile, $\tilde{Z}_y - Z_y = \int p(y|x_0)(\Phi_\# \gamma(x_0) - p_{\text{data}}(x_0)) dx_0$, and then

$$\begin{aligned} \frac{|\tilde{Z}_y - Z_y|}{Z_y} &\le \int \frac{1}{Z_y} p(y|x_0) |\Phi_\# \gamma(x_0) - p_{\text{data}}(x_0)| dx_0 \\ &\le \int \kappa_y |\Phi_\# \gamma(x_0) - p_{\text{data}}(x_0)| dx_0. \end{aligned}$$

Putting back to (A.1), we have

$$2\,\mathrm{TV}(p_{0,y}, \tilde{p}_{0,y}) \le 2\kappa_y \int \left| p_{\text{data}}(x_0) - \Phi_\# \gamma(x_0) \right| dx_0 = 4\kappa_y\, \mathrm{TV}(p_{\text{data}}, \Phi_\# \gamma),$$

which proves the theorem under (9). □

Proof of Lemma 4.2. By that $\tilde{p}_{0,y} = \Phi_\# \tilde{p}_{1,y}$, $\tilde{p}^S_{0,y} = \Phi_\# \tilde{p}^S_{1,y}$, and Data Processing Inequality. □

Proof of Corollary 4.3. By Theorem 4.1, Lemma 4.2, and triangle inequality since TV is half of the L^1 norm between two densities. □

B. Experimental Details

B.1. Details of the proposed approach

Consistency model generative process. To represent the map Φ from noise space to data space, we utilize the pre-trained CMs of [51] with a 1- or 2-step sampler. For the 2-step sampler, we use standard multistep consistency sampling (Algorithm 1, [51]), i.e.,

$$x_0 = f_\theta\left(f_\theta(x_T, T) + \sqrt{t^2 - \epsilon^2}z, t\right),$$

where f_θ is the pre-trained CM, $x_T \leftarrow x_1$, $T = 80$, $\epsilon = 2 \times 10^{-3}$ is a small noise offset , and t is an intermediate "time step" along the PF-ODE trajectory (the "halfway" point). In [51], z is sampled from the standard Gaussian for each call to Φ. In this work, we sample z once and fix it for all future calls to Φ, which we observe to empirically improve performance.

Warm-start initialization and sampling. The posterior sampling process begins with a warm-start initialization consisting of K steps of Adam optimization with learning rate, β_1, and β_2 for each experiment outlined in Tables A.1, A.2, and A.3. This is followed by N steps of Langevin dynamics simulation (via EM discretization in the main-text experiments) using step size τ. The NFEs per sample can be computed as $\eta(K + N)/N$, where η is the number of steps used for CM generation. All experiments are implemented in PyTorch and are run on a system with NVIDIA A100 GPUs.

See below for a pseudo-code implementation of one iteration of our sampling procedure:

```
x1_i = x1_i.requires_grad_()
x0_i = denoise(x1_i)

L = 1 / (2*sigma**2) * torch.norm(y - A(x0_i)) ** 2
g_i = torch.autograd.grad(outputs=L, inputs=x1_i)[0]

x1_i = x1_i - tau * (x1_i + g) + numpy.sqrt(2.*tau) * torch.randn_like(x1_i)
x1_i = x1_i.detach_()
```

Table A.1. Hyper-parameters for linear and nonlinear image restoration tasks on LSUN-Bedroom (256 x 256).

Method	8x Super-resolution	Gaussian Deblur	10% Inpainting	Nonlinear Deblur	Phase Retrieval	HDR Reconstruction
DPS-DM	$\zeta = 25, N = 100$	$\zeta = 7, N = 100$	$\zeta = 25, N = 100$	$\zeta = 15, N = 100$	$\zeta = 10, N = 100$	$\zeta = 5, N = 100$
MPGD-DM	$\zeta = 25, N = 100$	$\zeta = 15, N = 100$	$\zeta = 25, N = 100$	$\zeta = 7, N = 100$	$\zeta = 1, N = 100$	$\zeta = 5, N = 100$
LGD-DM	$\zeta = 25, M = 1, N = 100$	$\zeta = 25, M = 10, N = 100$	$\zeta = 7, M = 25, N = 100$	$\zeta = 9, M = 10, N = 100$	$\zeta = 1, M = 10, N = 100$	$\zeta = 30, M = 10, N = 100$
DPS-CM	$\zeta = 25, N = 100$	$\zeta = 7, N = 100$	$\zeta = 25, N = 100$	$\zeta = 8, N = 100$	$\zeta = 9, N = 100$	$\zeta = 4, N = 100$
MPGD-CM	N/A	N/A	N/A	$\zeta = 15, N = 100$	$\zeta = 3, N = 100$	$\zeta = 30, N = 100$
LGD-CM	$\zeta = 25, M = 1, N = 100$	$\zeta = 7, M = 1, N = 100$	$\zeta = 5, M = 1, N = 100$	$\zeta = 15, M = 10, N = 100$	$\zeta = 0.5, M = 10, N = 100$	$\zeta = 15, M = 10, N = 100$
Ours(1-step)	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-5}$	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-6}$	$K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-5}$	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 5 \times 10^{-6}$	Adam: $K = 200, \text{lr} = 1 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-6}$	$K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-6}$
Ours(2-step)	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-5}$	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-7}$	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-5}$	Adam: $K = 500, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 5 \times 10^{-6}$	Adam: $K = 500, \text{lr} = 1 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-6}$	Adam: $K = 500, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-6}$

Table A.2. Hyper-parameters for linear image restoration tasks on ImageNet (64 x 64).

Method	4x Super-resolution	Gaussian Deblur	20% Inpainting
DPS-DM	$\zeta = 20, N = 100$	$\zeta = 15, N = 100$	$\zeta = 30, N = 100$
LGD-DM	$\zeta = 3, M = 10, N = 100$	$\zeta = 1, M = 10, N = 100$	$\zeta = 5, M = 10, N = 100$
DPS-CM	$\zeta = 30, N = 100$	$\zeta = 30, N = 100$	$\zeta = 25, N = 100$
LGD-CM	$\zeta = 3, M = 10, N = 100$	$\zeta = 7, M = 10, N = 100$	$\zeta = 6, M = 10, N = 100$
Ours(1-step)	Adam: $K = 800, \text{lr} = 1 \times 10^{-2}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 5 \times 10^{-4}$	Adam: $K = 800, \text{lr} = 1 \times 10^{-2}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 3 \times 10^{-5}$	$K = 800, \text{lr} = 1 \times 10^{-2}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-4}$
Ours(2-step)	Adam: $K = 500, \text{lr} = 5 \times 10^{-2}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-4}$	Adam: $K = 500, \text{lr} = 5 \times 10^{-2}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 3 \times 10^{-5}$	Adam: $K = 500, \text{lr} = 5 \times 10^{-2}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-4}$

B.2. Details of the baselines

The baseline methods conduct $t = 1, \ldots, N$ Euler steps for sampling. All methods require a denoiser to provide $x_0 \approx \hat{x}_0(x_t)$ at each sampling step t, which is achieved using either a pre-trained EDM [29] or CM [51], both obtained from [51] for each dataset.

Table A.3. Hyper-parameters for linear and nonlinear diversity experiments on LSUN-Bedroom (256 x 256).

Method	8x Super-resolution	Gaussian Deblur	10% Inpainting	Nonlinear Deblur	Phase Retrieval	HDR Reconstruction
DPS-DM	$\zeta = 7, N = 100$	$\zeta = 7, N = 100$	$\zeta = 7, N = 100$	$\zeta = 5, N = 100$	$\zeta = 5, N = 100$	$\zeta = 1, N = 100$
LGD-DM	$\zeta = 15, M = 1, N = 100$	$\zeta = 5, M = 1, N = 100$	$\zeta = 15, M = 1, N = 100$	$\zeta = 4, M = 10, N = 100$	$\zeta = 0.5, M = 10, N = 100$	$\zeta = 10, M = 10, N = 100$
Ours(1-step)	Adam: $K = 400, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 4 \times 10^{-4}$	Adam: $K = 600, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-6}$	$K = 600, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-4}$	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 25, \tau = 7.5 \times 10^{-6}$	Adam: $K = 200, \text{lr} = 1 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 25, \tau = 3 \times 10^{-6}$	$K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 25, \tau = 3 \times 10^{-6}$
Ours(2-step)	Adam: $K = 600, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 4 \times 10^{-4}$	Adam: $K = 600, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-5}$	Adam: $K = 800, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 10, \tau = 1 \times 10^{-4}$	Adam: $K = 500, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 25, \tau = 7.5 \times 10^{-6}$	Adam: $K = 500, \text{lr} = 1 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 25, \tau = 3 \times 10^{-6}$	Adam: $K = 500, \text{lr} = 5 \times 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.999$ EM: $N = 25, \tau = 3 \times 10^{-6}$

Diffusion Posterior Sampling (DPS). DPS [12] utilizes the denoiser corresponding to a pre-trained DM to approximate the measurement likelihood gradient at each step of DM sampling. At each state x_t along the diffusion sampling trajectory, a score-base diffusion model can provide a predicted $\hat{x}_0(x_t)$, which can be used to compute $\nabla_{x_t} p(y|\hat{x}_0)$ via differentiation through the score-based model. In DPS, each step of diffusion sampling is adjusted by this gradient with weight ζ, i.e., $x_{t-1} \leftarrow x_{t-1} - \zeta \nabla_{x_t} p(y|\hat{x}_0)$.

Manifold Preserving Guided Diffusion (MPGD). MPGD [20] computes the gradient of the measurement likelihood in the denoised space rather than with respect to x_t at each step, taking a gradient step in $\hat{x}_0$ before updating the diffusion iterate. That is, MPGD conducts the update $\hat{x}_0 \leftarrow \hat{x}_0(x_t) - \zeta \nabla_{\hat{x}_0} p(y|\hat{x}_0(x_t))$, which can then be use to yield x_{t-1} at each step. MPGD also provides an optional manifold projection step which utilizes pre-trained autoencoders to ensure $\hat{x}_0$ remains on the data manifold. For a fair comparison, we only consider MPGD without manifold projection in this work.

Loss Guided Diffusion (LGD). LGD [49] aims to improve the approximation of $p(y|x_0)$ at each step along the sampling trajectory via a Monte Carlo approach. Viewing $p(y|\hat{x}_0)$ in DPS as a delta distribution approximation of $p(y|x_0)$ about $\hat{x}_0$, LGD instead computes the log-mean-exponential of $p(y|\hat{x}_0^{(m)})$ for $m = 1, \ldots, M$ perturbed copies of $\hat{x}_0$. That is, $p(\hat{x}_0|x_t) \sim \mathcal{N}(\hat{x}_0(x_t), r_t^2 I)$, where $r_t = \beta_t / \sqrt{1 + \beta_t^2}$. The weighted (by ζ) Monte Carlo gradient $\nabla_{x_t} \log \left(\frac{1}{M} \sum_{m=1}^{M} \exp \left(p \left(y|\hat{x}_0^{(m)} \right) \right) \right)$ is then used to adjust x_{t-1}, as in DPS.

B.3. Degradations and forward operators

In all experiments, pixel values are scaled from [-1, 1] (as in [51]) before application of forward operators. The details of the measurement likelihoods corresponding to each forward operator are outlined below. All methods use $\sigma = 0.1$, except for phase retrieval, which uses $\sigma = 0.05$.

Super-resolution. The super-resolution task is defined by the following measurement likelihood:

$$y \sim \mathcal{N}(y|\text{AvgPool}_f(x), \sigma^2 I),$$

where AvgPool represents 2D average pooling by a factor f.

Gaussian deblur. Gaussian blur is defined by a block Hankel matrix C^{ψ} representing convolution of x with kernel ψ:

$$y \sim \mathcal{N}(y|C^{\psi} x, \sigma^2 I).$$

We consider a 61 x 61 Gaussian kernel with standard deviation of 3.0, as in [12].

Inpainting. The measurement likelihood corresponding to $p\%$ inpainting is a function of a mask P with (1-p)% uniformly random 0 values:

$$y \sim \mathcal{N}(y|Px, \sigma^2 I).$$

Nonlinear deblur. Following [12], the forward nonlinear blur operator is a pre-trained neural network $\mathcal{F}_{\phi}$ to approximate the integration of non-blurry images over a short time frame given a single sharp image [52]. Therefore, the measurement likelihood is as follows:

$$y \sim \mathcal{N}(y|\mathcal{F}_{\phi}(x), \sigma^2 I).$$

Phase retrieval. The forward operator of the phase retrieval task takes the absolute value of the 2D Discrete Fourier Transform F applied to x: $|Fx|$. However, since this task is known to be highly ill-posed [12, 19], an oversampling matrix P is also applied (with oversampling ratio 1 in this work):

$$y \sim \mathcal{N}(y||FPx|, \sigma^2 I).$$

High dynamic range reconstruction. In the HDR forward model, pixel values are scaled by a factor of 2 before truncation back to the range [-1, 1]. Therefore, the measurement likelihood is as follows:

$$y \sim \mathcal{N}(y|\text{clip}(2x, -1, 1), \sigma^2 I),$$

where $\text{clip}(\cdot, -1, 1)$ truncates all input values to the range [-1, 1].

C. Additional experiments

Numerical SDE solver comparison. Alternative numerical methods to EM (11) can be applied to discretize the Langevin dynamics SDE, such as the exponential integrator (EI) [24]. The EI scheme discretizes the nonlinear drift term $g^i = \nabla_{x_1} L_y(x_0)|_{x_1=z^i}$ and integrates the continuous-time dynamics arising from the linear term:

$$z^{i+1} = e^{-\tau} z^i - (1 - e^{-\tau}) g^i + \sqrt{1 - e^{-2\tau}} \xi^i,$$

where $\xi^i \sim \mathcal{N}(0, I)$. In Table A.4, quantitative comparison between our method using EM versus EI is shown on generating 10 samples for 100 images from the LSUN-Bedroom validation dataset, where the forward operator is nonlinear blurring. The same hyper-parameters are used for both methods, which are outlined in Table A.1. In this case, there is a marginal improvement in most metrics when using the EI scheme.

Table A.4. Comparison between our method with EM and EI integration on the nonlinear deblur task on LSUN-Bedroom (256 x 256).

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Ours-EM(1-step)	20.3	0.566	0.440	76.7
Ours-EM(2-step)	18.7	0.501	0.492	73.3
Ours-EI(1-step)	20.5	0.569	0.437	76.3
Ours-EI(2-step)	18.7	0.504	0.491	74.2

D. Additional qualitative results

Visualizations of additional reconstructions from our method corresponding to the linear and nonlinear experiments from Section 6.1 can be found in Figures A.2, A.3, A.4, A.5, and A.6. Additionally, diverse sets of samples from our one-step/two-step CM method corresponding to the experiments of Section 6.2 are visualized in Figures A.7, A.8, A.9, A.10, A.11, and A.12. Finally, diverse samples via the linear tasks on ImageNet (64 x 64) are shown in Figures A.13, A.14, and A.15. In these experiments, we use the one-step CM sampler with the same hyper-parameters as in Table A.2, but with $\tau = 4 \times 10^{-4}$ for inpainting, $\tau = 9 \times 10^{-4}$ for super-resolution, and $\tau = 5 \times 10^{-5}$ for Gaussian deblur.

E. Further Methodological Details

Inference time and sampling efficiency. In Figure A.1, we report a time comparison between sample accumulation via DPS-DM and our method with 1-step CM sampling, including 500 steps of warm start for our method (see Section 5). While there is a large upfront NFE and time cost for warm start, this cost is amortized over sample accumulation. Each additional sample of DPS requires 100 NFEs, while each additional sample via our method is acquired in a single function evaluation. This corresponds to a much more gradual increase in overall computation time as more samples are generated.

Fixed-noise consistency model sampling. We utilize a fixed noise code $z \sim \mathcal{N}(0, I)$ for multi-step CM sampling, as described in Section B.1. Typical CM sampling utilizes distinct random code(s) $z \sim \mathcal{N}(0, I)$ each time Φ is called. However, this results in stochastic sampler Φ, violating the interpretation of Φ as a pushforward map, rendering the theoretical guarantee of posterior sampling invalid. Moreover, we observe that using stochastic Φ in practice (i.e., distinct z) leads to instability in the sampling scheme and less consistent samples.

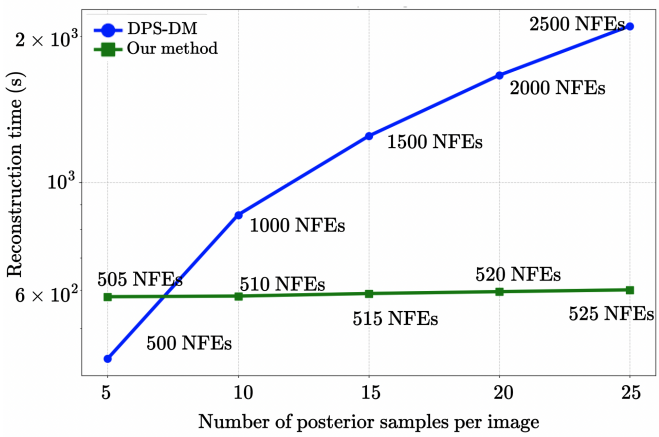

Figure A.1. Reconstruction time comparison between DPS-DM and our method for varying numbers of posterior samples. DPS-DM scales poorly with the number of samples, while our method maintains a nearly constant time, demonstrating significantly lower computational cost. The corresponding Number of Function Evaluations (NFEs) (including NFEs for the warmup stage) values per image are annotated.

Table A.5. Fidelity/diversity trade-off for 8× SR (left) and nonlinear deblur (right).

τ	1e−5	2.1e−5	4.4e−5	9.1e−5	1.9e−5	4e−5	τ	5e-6	5.5e-06	6e-6	6.5e-6	7e-6	7.5e-6
PSNR ↑	19.9	20.0	20.0	19.8	19.1	15.7	PSNR ↑	20.2	19.7	18.1	17.4	16.5	15.4
SSIM ↑	0.515	0.514	0.514	0.510	0.495	0.442	SSIM ↑	0.548	0.541	0.515	0.522	0.506	0.481
LPIPS ↓	0.433	0.432	0.432	0.435	0.448	0.496	LPIPS ↓	0.449	0.454	0.482	0.479	0.497	0.516
FID ↓	87.7	88.1	87.6	88.0	87.5	90.5	FID ↓	94.7	97.6	105	99.4	98.4	106
DS ↑	2.72	2.53	2.52	2.90	3.39	2.78	DS ↑	3.28	3.09	4.12	2.85	2.56	2.71
CS ↓	0.999	0.998	0.997	0.992	0.973	0.918	CS ↓	0.964	0.958	0.940	0.934	0.921	0.908

One- and two-step consistency model sampling. In this work, we conduct single- and multi-step CM sampling [51]; see Section 5 and Section B.1. Experimentally, we observe that one-step sampling results in higher-fidelity posterior samples (Tables 1 and 2) while the two-step sampling results in slightly more diverse samples (Table 3). Our interpretation of these results is as follows: Two main factors explain this. (1) Absence of Speed-Quality trade-off in CM: Recent work [32] indicates that CMs lack the usual speed–quality trade-off, as multi-step intervals overlap and do not necessarily improve results. This occurs due to overlapping time intervals ("jumps") in multi-step samplers, and Theorem 1 in [32] provides a formal explanation. (2) Fixed noise in multi-step sampling: Our 2-step sampler (Supp. B.1) fixes injected noise z at each step to avoid divergence that can arise from using fresh noise at every iteration. While this provides stability, it may also remove beneficial stochasticity, diminishing performance gains expected from multi-step approaches.

Details of base models. For the experiments in Section 6, we have used EDMs and distilled CMs reported in the original CM work [51]. In unconditional generation, EDM achieves FID of 3.57 on LSUN-Bedroom (256 × 256) and 2.44 on ImageNet (64 × 64) with 79-step sampling. The corresponding distilled CMs achieve FID of 5.22 and 4.70 for 2-step sampling in LSUN-Bedroom (256 × 256) and ImageNet (64 × 64), respectively. Due to the gap in fidelity between unconditional samples drawn using the diffusion models and consistency models, it is unfair to compare posterior samples between the two model types. For this reason, we introduce the CM-based baselines in Section 6, to ensure a more fair comparison.

Advantages and disadvantages of the proposed approach. Our key innovation lies in performing Langevin sampling directly in the noise space of a pre-trained generative model. Unlike methods that optimize for a point estimate, our approach combines a warm-start step with posterior sampling dynamics, theoretically approximating samples from the posterior distribution. This enables the generation of multiple diverse reconstructions with competitive costs compared to optimization-based baselines. A notable advantage of our approach is that it avoids mode collapse and posterior over-regularization often observed in GANs and VAEs, leveraging the strong prior modeling capacity of consistency models (CMs). Unlike GANs, which require adversarial training, or VAEs, which may suffer from latent bottlenecks, CMs enable efficient, stable sampling with strong expressiveness.

Limitations. A potential limitation is the reliance on a well-behaved generative model with deterministic mapping, which may restrict flexibility in scenarios involving multimodal or highly stochastic observations. Additionally, although our method is efficient, the warm-start step adds an extra layer of computation compared to naive sampling.

F. Additional Related Works

Energy-based models and noise space sampling. Using MCMC to accumulate samples is related to EBMs. Early works used convolutional EBMs to conduct Langevin sampling from a natural image starting point [56] or a synthetic initialization [57]. Other works have utilized generative models' noise space for EBM sampling [25, 40, 55, 61]. For instance, [17] utilize flows to improve noise constrastive estimation of EBMs. Similarly, [61] define the latent noise space of generative models by an EBM, upon which they learn a saliency map model. [25, 40] conduct HMC sampling in the noise space of flows; similar approaches adapted variational autoencoders for EBM sampling [55]. Such approaches facilitate efficient sampling by leveraging tractability in the noise space. Similarly, our method simulates Langevin dynamics in a tractable noise space, but we focus on the posterior sampling task as opposed to unconditional sampling.

G. Ablation Study

Fidelity vs diversity trade-off. We analyze the effect of the step size τ on fidelity and diversity across 8× SR and nonlinear deblurring tasks (Table A.5). Larger values of τ promote faster exploration in noise space, resulting in higher diversity—as reflected by increasing Diversity Score (DS) and decreasing CLIP Similarity (CS). For example, in 8× SR, DS increases from 2.52 to 3.39 as τ grows from 4.4×10^{-5} to 1.9×10^{-5}, while CS drops from 0.997 to 0.973. Similarly, for nonlinear deblurring, DS peaks at 4.12 for $\tau = 6 \times 10^{-6}$, indicating strong diversity. Notably, this gain in diversity comes with only marginal degradation in fidelity. For instance, in SR, PSNR drops slightly (from 20.0 to 19.1) and LPIPS rises marginally (from 0.432 to 0.448). These results demonstrate that our method can generate diverse posterior samples with minimal compromise in reconstruction quality.

References

[1] Muhammad Asim, Max Daniels, Oscar Leong, Ali Ahmed, and Paul Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In *ICML*, 2020. 8

[2] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. In *ICML*, 2024. 8

[3] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *ICML*, 2018. 8

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 1

[5] Joan Bruna and Jiequn Han. Posterior sampling with denoising oracles via tilted transport. Available online : https://arxiv.org/abs/2407.00745, 2024. 6, 8

[6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018. 2, 4, 8

[7] Sitan Chen, Giannis Daras, and Alex Dimakis. Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers. In *ICML*, 2023. 3

[8] Xiuyuan Cheng, Jianfeng Lu, Yixin Tan, and Yao Xie. Convergence of flow-based generative models via proximal gradient descent in Wasserstein space. *IEEE Transactions on Information Theory*, 2024. 3

[9] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, 2021. 1, 6

[10] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *CVPR*, 2022. 1, 6

[11] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *NeurIPS*, 2022. 1, 6

[12] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023. 1, 2, 5, 6, 3, 4

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[14] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 1

[15] Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *ICLR*, 2024. 6

[16] Federico Galatolo., Mario Cimino., and Gigliola Vaglini. Generating images from caption and vice versa via clip-guided generative latent space search. In *International Conference on Image Processing and Vision Engineering*, 2021. 8

[17] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *CVPR*, 2020. 6

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 1, 8

[19] Monson Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(2):140–154, 1982. 4

[20] Linchao He, Hongyu Yan, Mengting Luo, , Hongjie Wu, Kunming Luo, Wang Wang, Wenchao Du, Hu Chen, Hongyu Yang, Yi Zhang, and Jiancheng Lv. Fast and stable diffusion inverse solver with history gradient update. Available online : https://arxiv.org/pdf/2307.12070, 2023. 3

[21] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *ICLR*, 2024. 1, 5, 6

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 1, 2

[24] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010. 4

[25] Matthew Hoffman, Pavel Sountsov, Joshua V Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport. Available online: https://arxiv.org/abs/1903.03704, 2019. 2, 6

[26] Daniel Zhengyu Huang, Jiaoyang Huang, and Zhengjiang Lin. Convergence analysis of probability flow ode for score-based generative models. Available online: https://arxiv.org/abs/2404.09730, 2024. 3

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1

[29] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 2, 5

[30] Bahjat Kawar, Gregory Vaksman, and Michael Elad. SNIPS: Solving noisy inverse problems stochastically. In *NeurIPS*, 2021. 1, 6

[31] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *NeurIPS*, 2022. 1, 6

[32] Kim et al. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *ICLR*, 2024. 5

[33] Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. In *ICLR*, 2024. 3

[34] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. SRDiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 1, 6

[35] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 2

[36] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2SB: Image-to-Image Schrödinger bridge. In *ICML*, 2023. 6

[37] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 2

[38] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 1, 6

[39] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. Available online : https://arxiv.org/abs/1411.1784, 2014. 1

[40] Erik Nijkamp, Ruiqi Gao, Pavel Sountsov, Srinivas Vasudevan, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Mcmc should mix: Learning energy-based model with neural transport latent space mcmc. In *ICLR*, 2022. 2, 6

[41] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 8

[42] Ashwini Pokle, Matthew J Muckley, Ricky TQ Chen, and Brian Karrer. Training-free linear image inversion via flows. *TMLR*, 2023. 8

[43] Vishal Purohit, Junjie Luo, Yiheng Chi, Qi Guo, Stanley H. Chan, and Qiang Qiu. Generative Quanta Color Imaging. In *CVPR*, 2024. 1

[44] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH*, 2022. 1, 6

[45] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022. 6

[46] Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Conditional simulation using diffusion schrödinger bridges. In *UAI*, 2022. 1, 6

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2, 6

[48] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *ICLR*, 2023. 1, 6, 8

[49] Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *ICML*, 2023. 1, 5, 6, 3

[50] 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. 2, 3

[51] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023. 2, 5, 8, 3

[52] Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *CVPR*, 2021. 5, 3

[53] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *ICLR*, 2023. 1, 6

[54] Jay Whang, Qi Lei, and Alex Dimakis. Solving inverse problems with a flow-based noise model. In *ICML*, 2021. 8

[55] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. VAEBM: a symbiosis between variational autoencoders and energy-based models. In *ICLR*, 2021. 2, 6

[56] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *ICML*, 2016. 2, 6

[57] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2, 6

[58] Chen Xu, Xiuyuan Cheng, and Yao Xie. Normalizing flow neural networks by JKO scheme. In *NeurIPS*, 2023. 3

[59] Xingyu Xu and Yuejie Chi. Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction. In *NeurIPS*, 2024. 6

[60] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. Available online : https://arxiv.org/abs/1506.03365, 2024. 5

[61] Jing Zhang, Jianwen Xie, Nick Barnes, and Ping Li. Learning generative vision transformer with energy-based latent space for saliency prediction. *NeurIPS*, 2021. 6

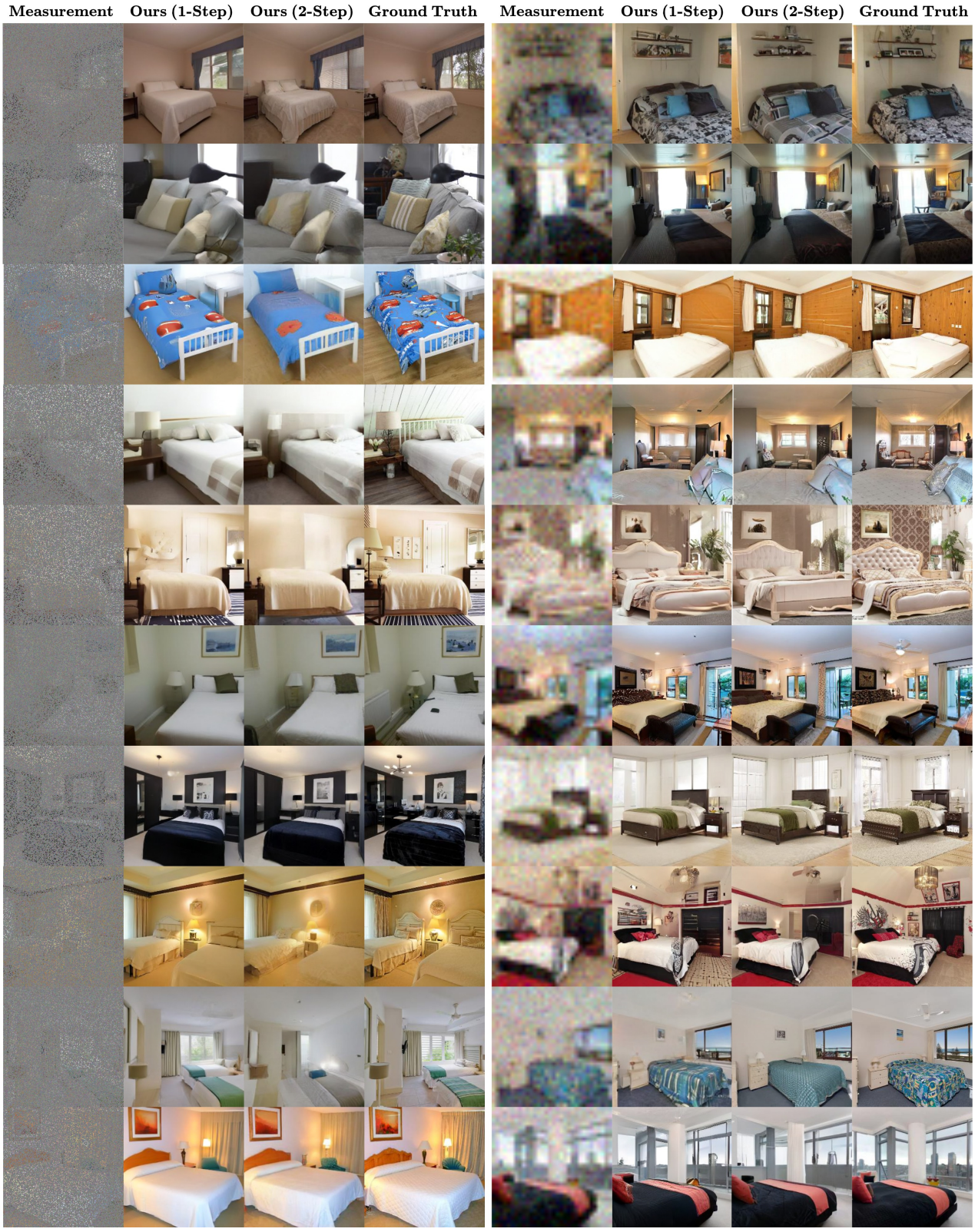

Figure A.2. Additional image reconstructions for inpainting (left) and 8x super-resolution (right) on LSUN-Bedroom (256 x 256).

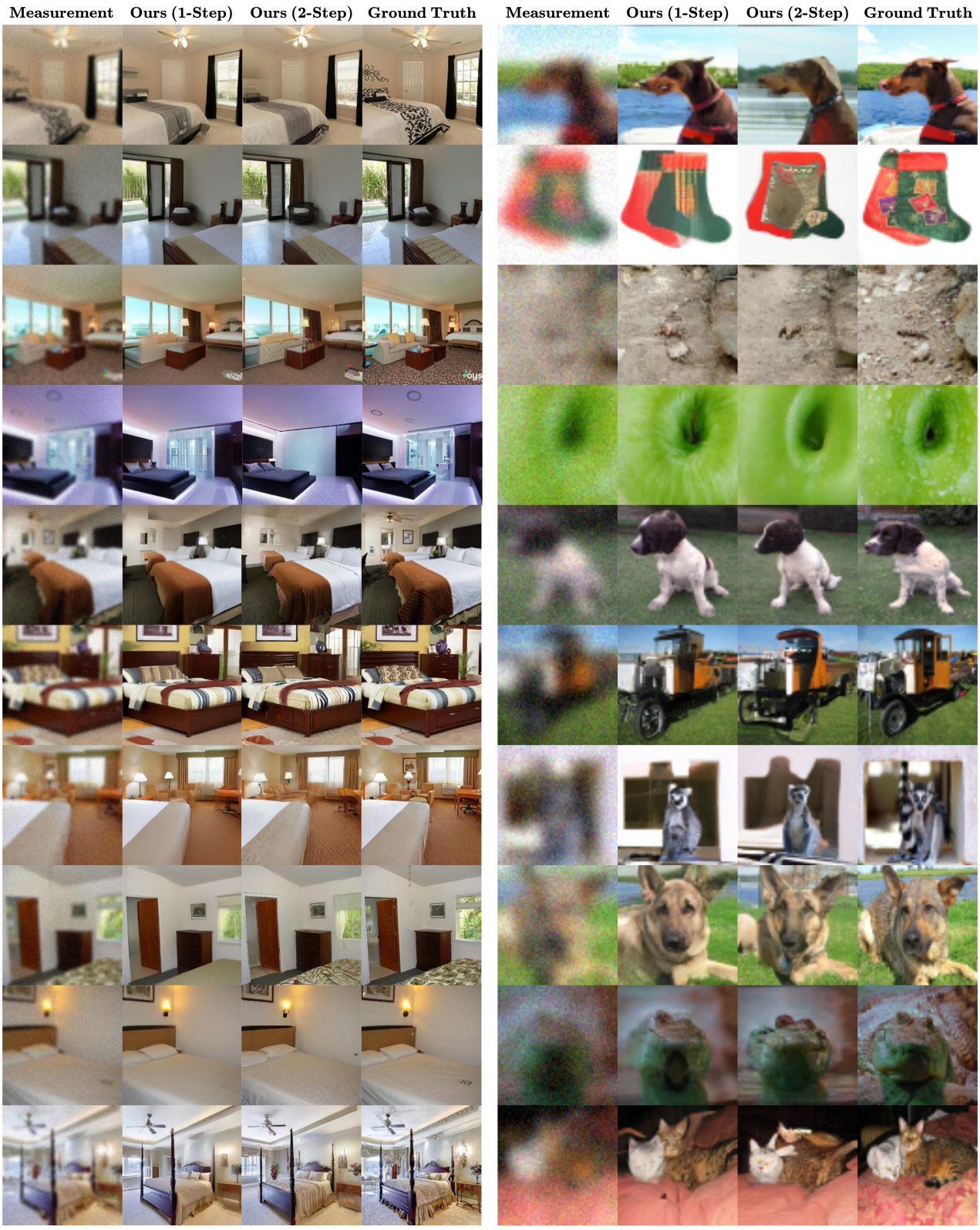

Figure A.3. Additional image reconstructions for Gaussian Deblurring on LSUN-Bedroom (256 x 256) (left) and ImageNet (64 x 64) (right).

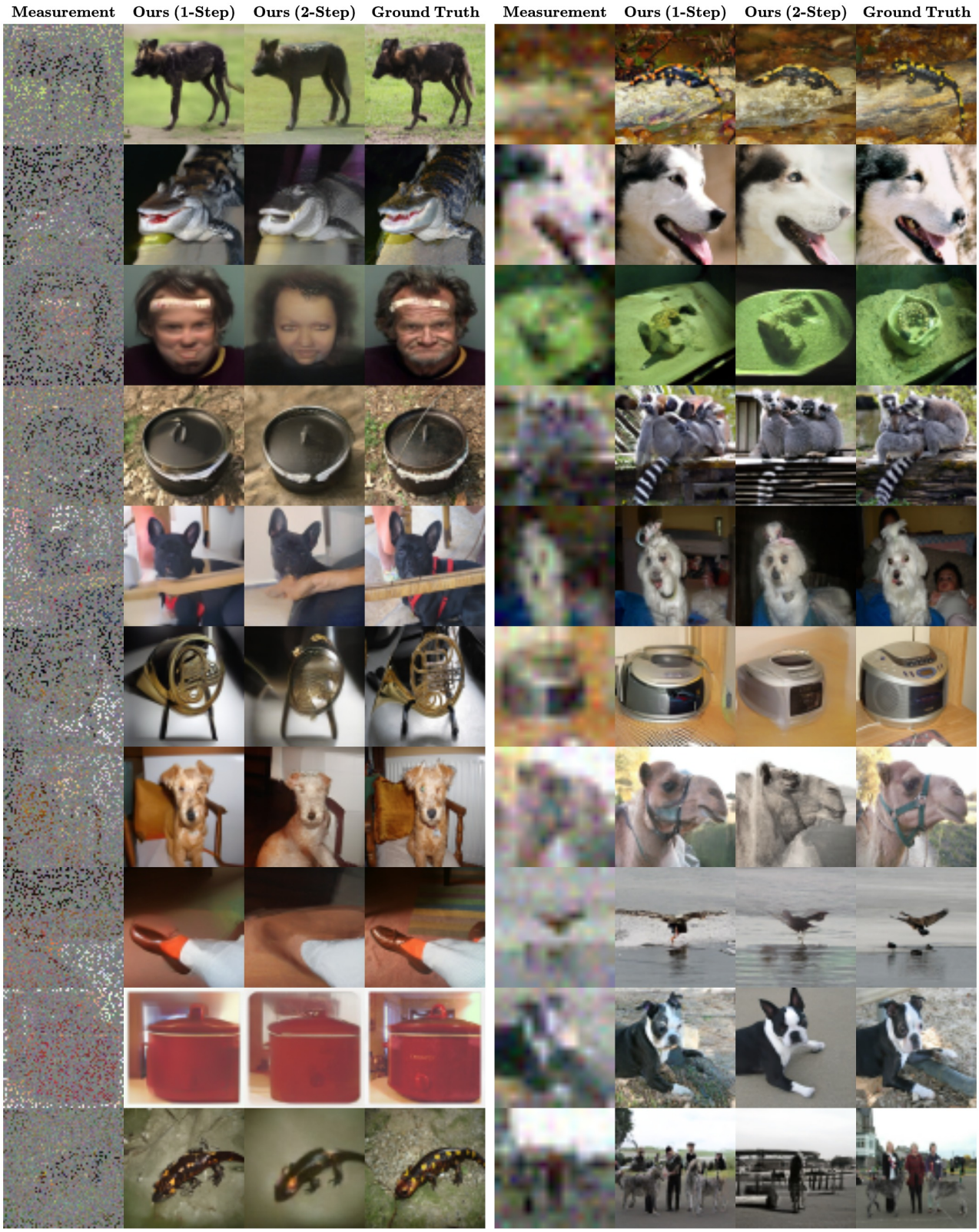

Figure A.4. Additional image reconstructions for inpainting (left) and 4x super-resolution (right) on ImageNet (64 x 64).

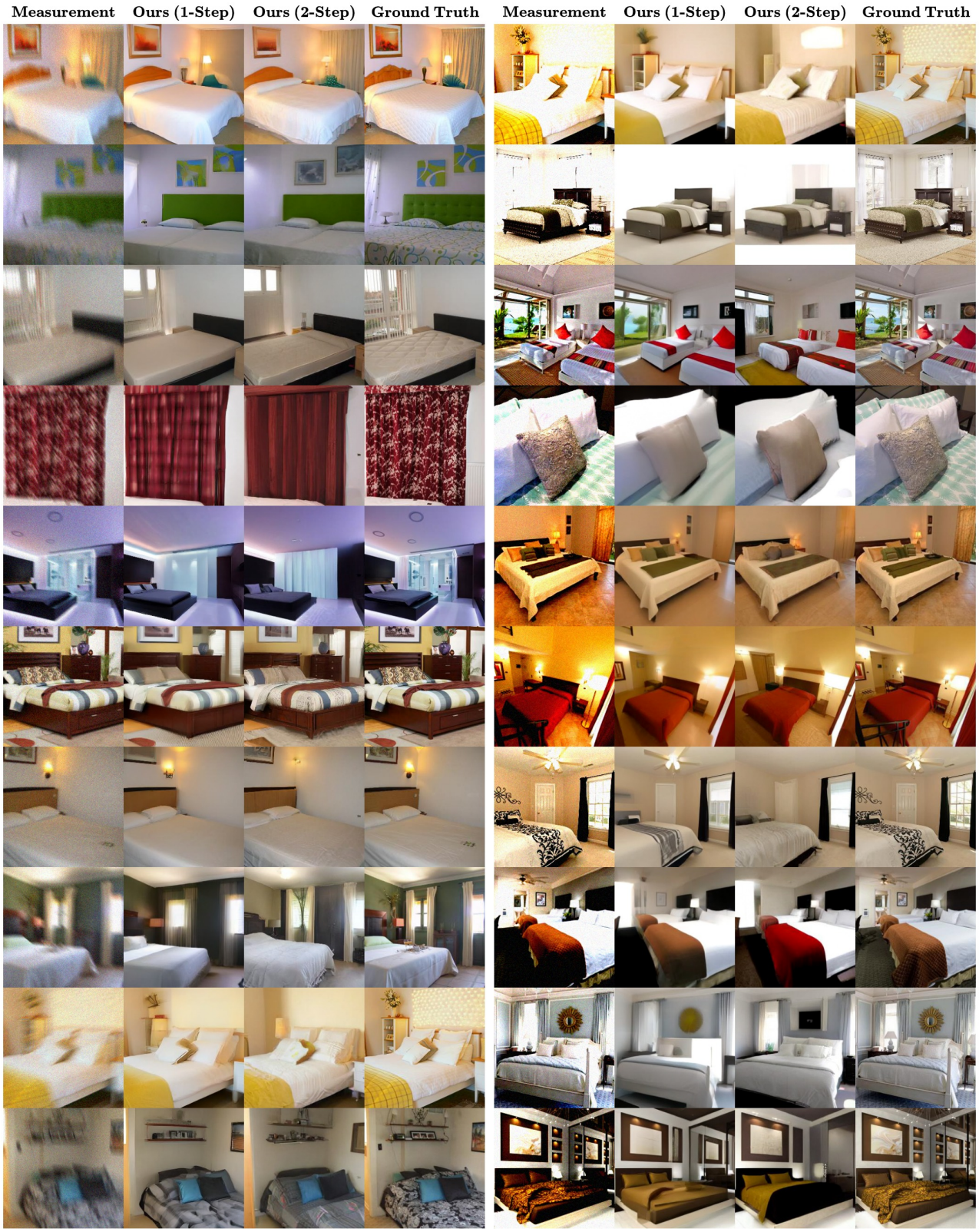

Figure A.5. Additional image reconstructions for nonlinear deblur (left) and HDR reconstruction (right) on LSUN-Bedroom (256 x 256).

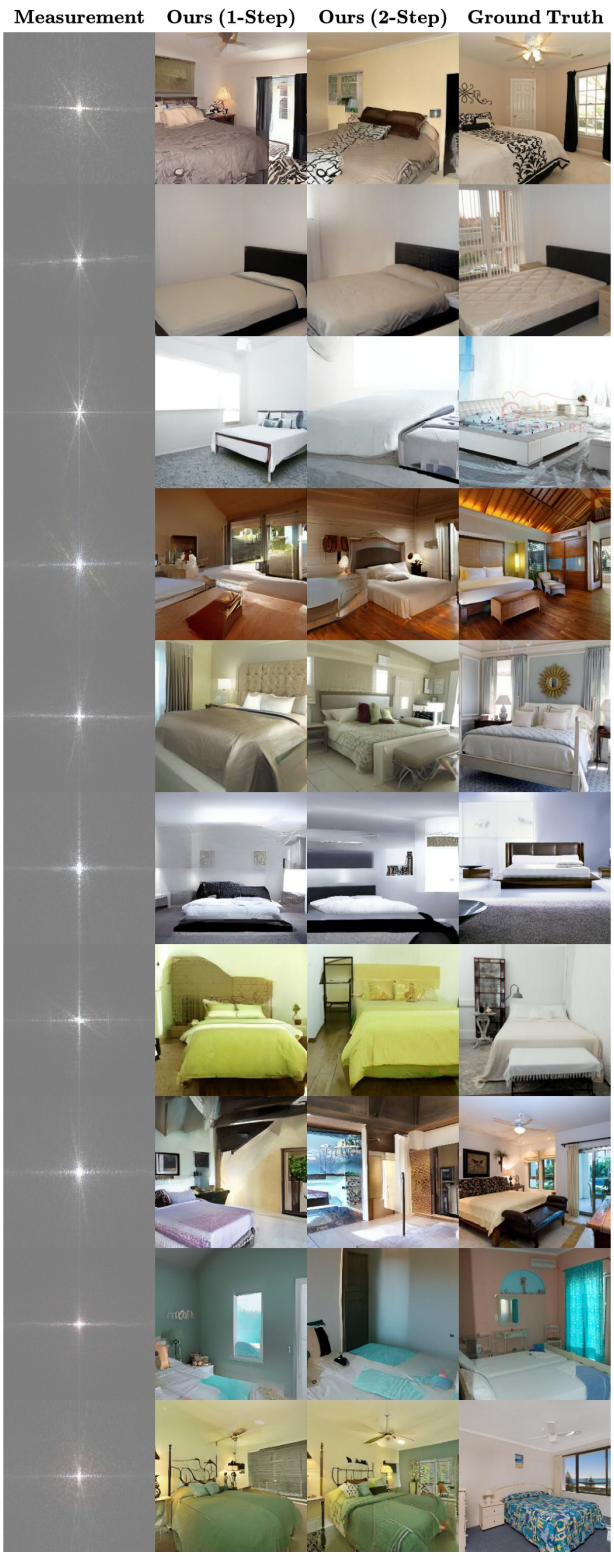

Figure A.6. Additional image reconstructions for phase retrieval on LSUN-Bedroom (256 x 256).

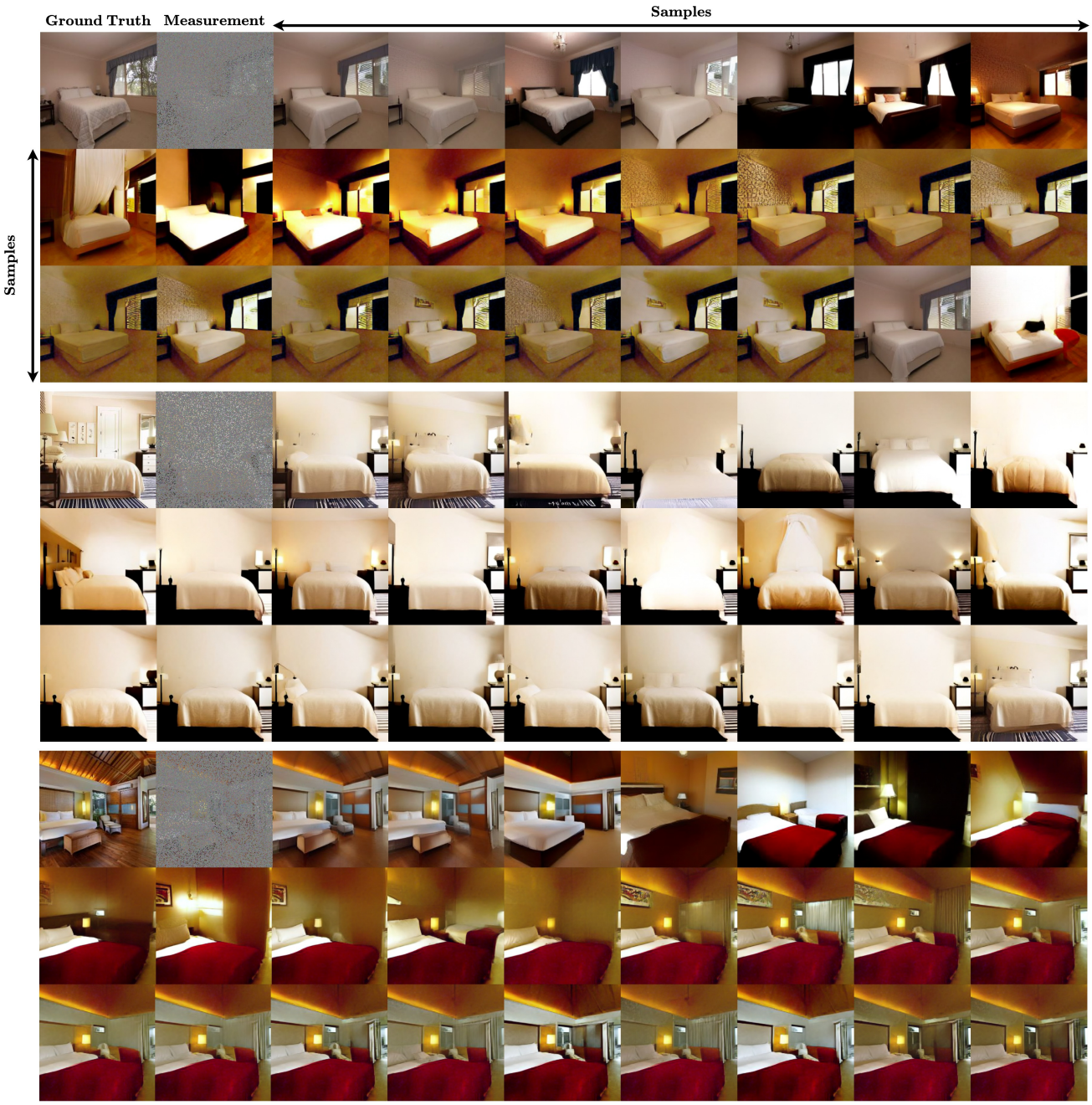

Figure A.7. Additional sets of samples for Inpainting (10%) on LSUN-Bedroom (256 x 256).

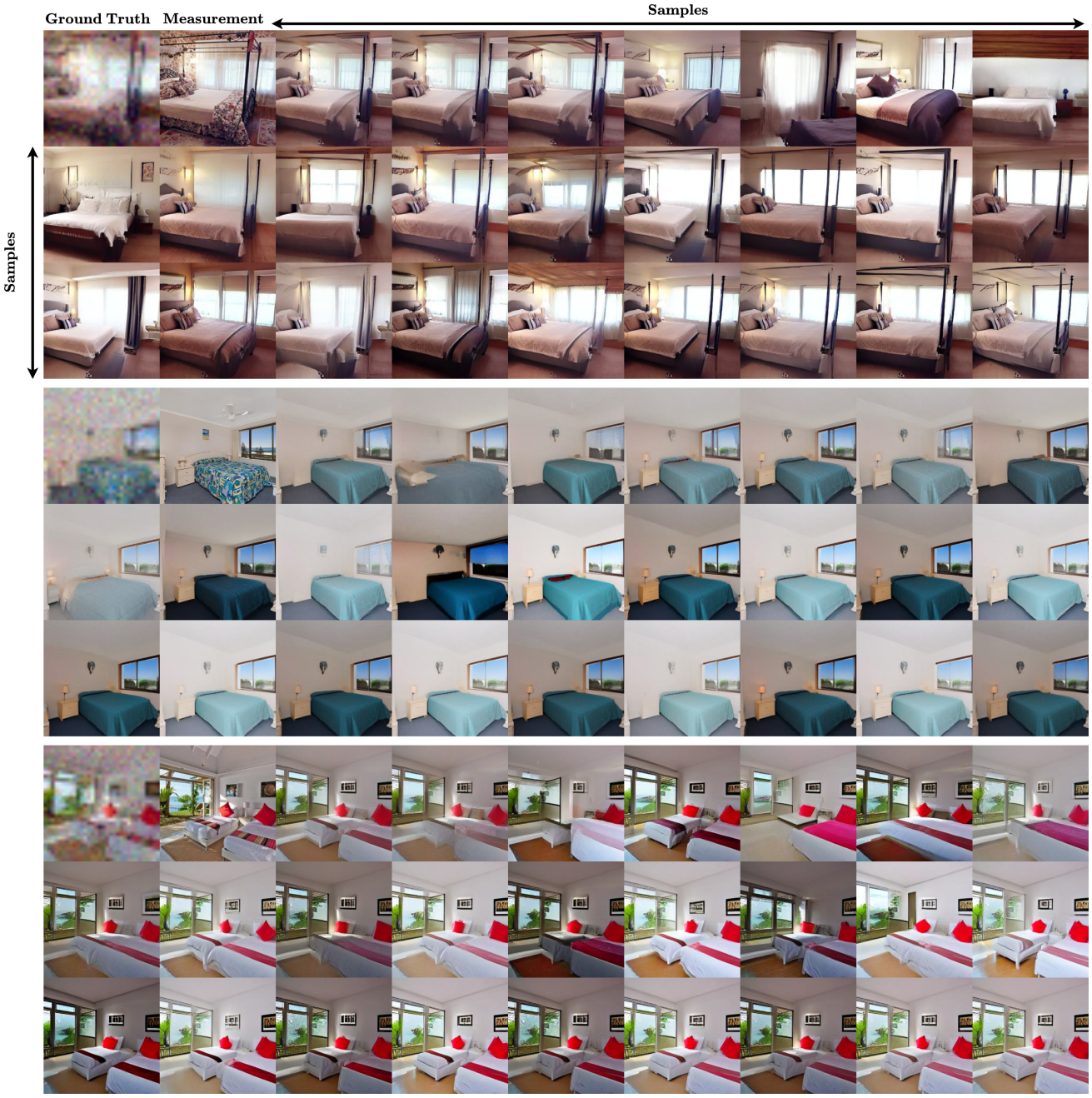

Figure A.8. Additional sets of samples for SR (8x) on LSUN-Bedroom (256 x 256).

Figure A.9. Additional sets of samples for SR (8x) on LSUN-Bedroom (256 x 256) for 2-step method.

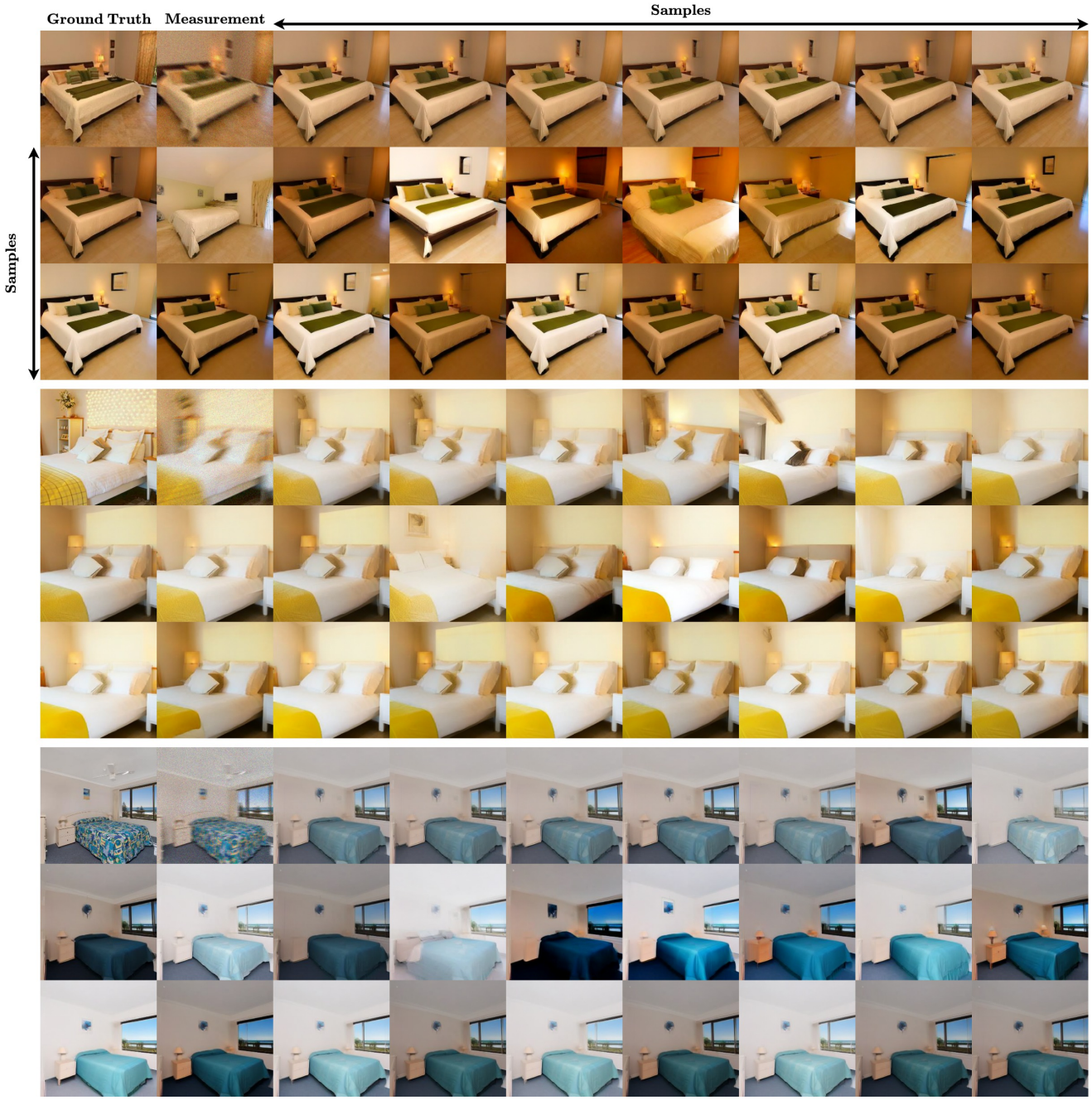

Figure A.10. Additional sets of samples for nonlinear deblur on LSUN-Bedroom (256 x 256).

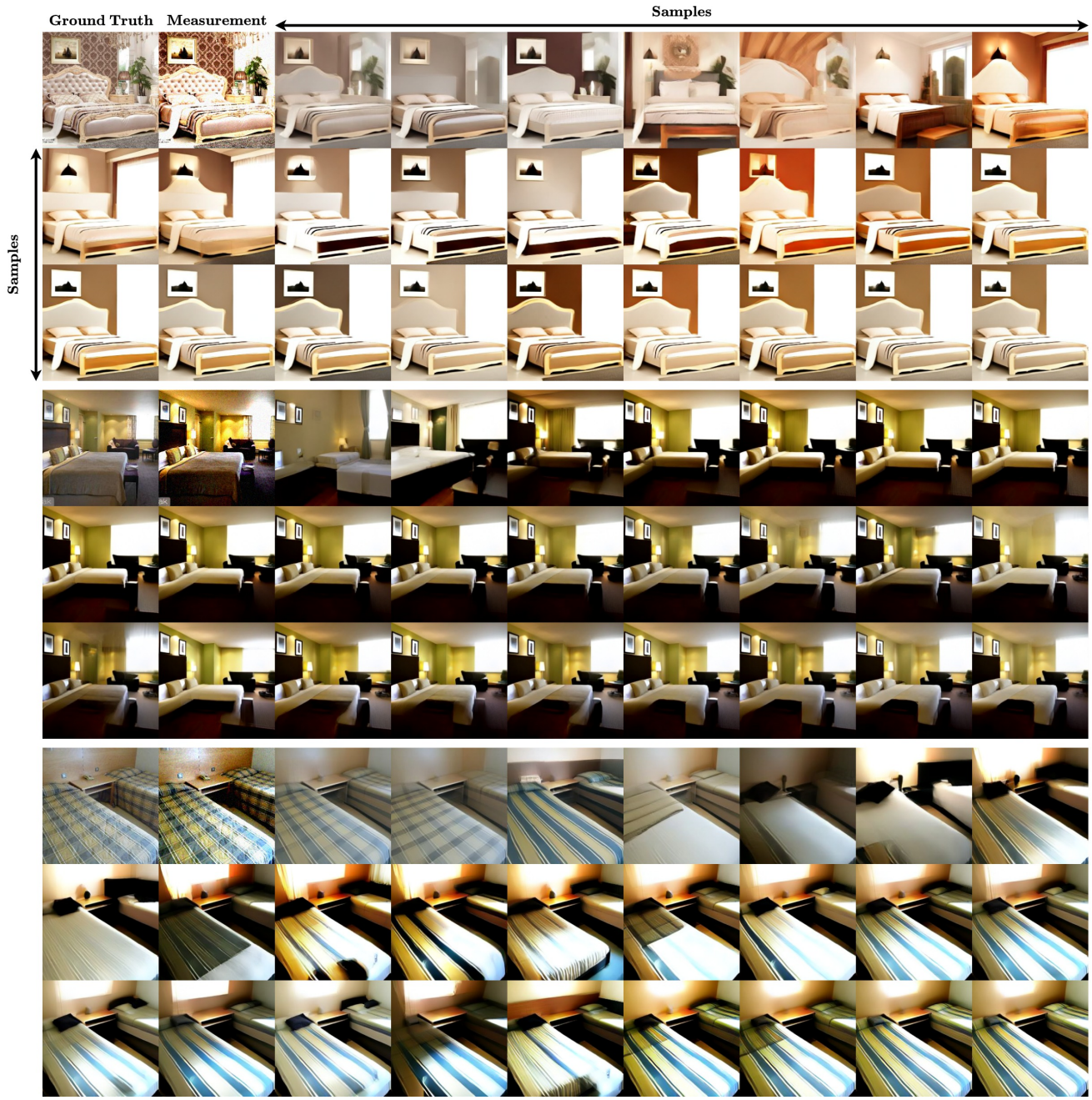

Figure A.11. Additional sets of samples for HDR reconstruction on LSUN-Bedroom (256 x 256).

Figure A.12. Additional sets of samples for phase retrieval on LSUN-Bedroom (256 x 256).

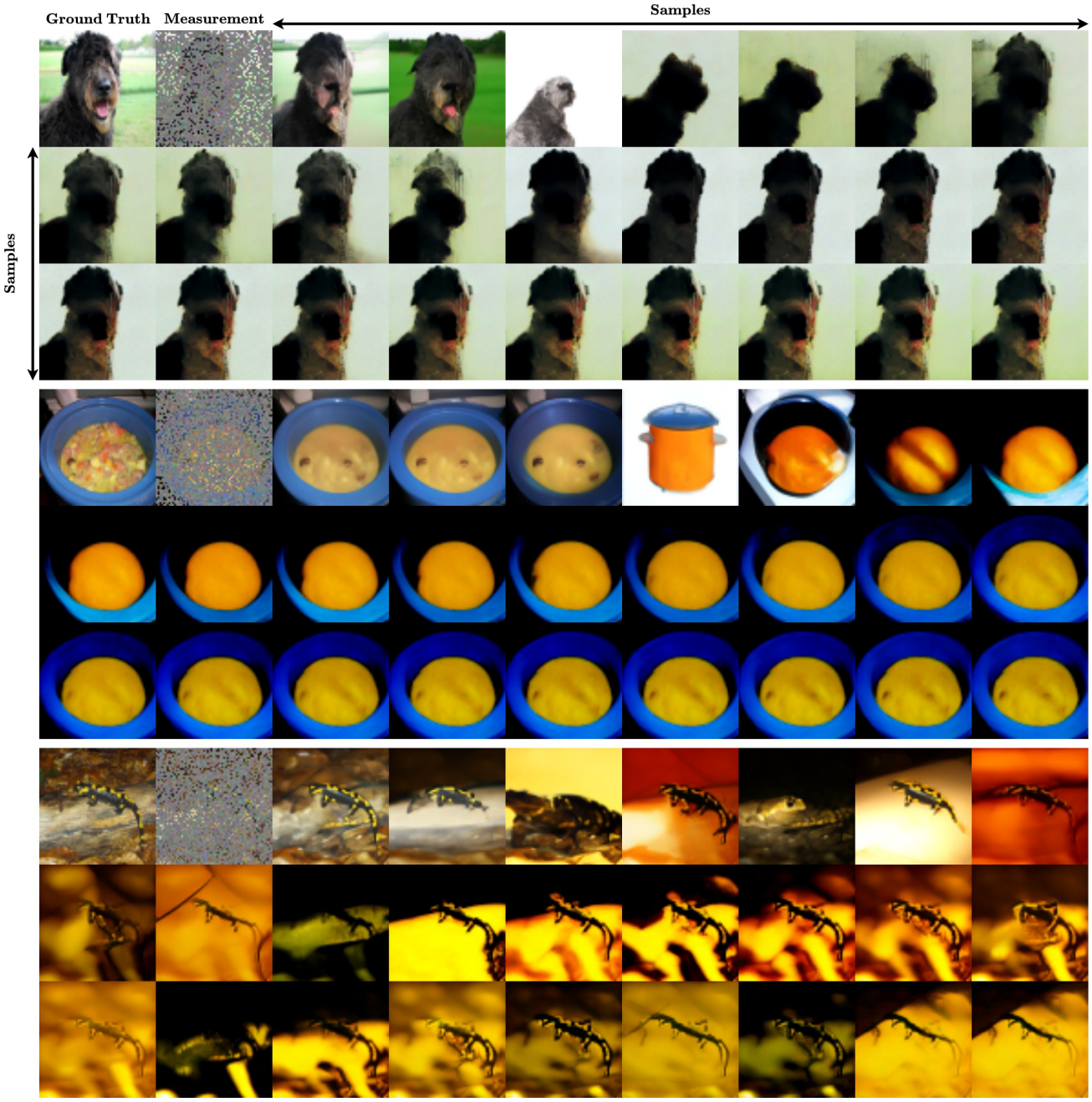

Figure A.13. Sets of samples for 20% inpainting on ImageNet (64 x 64).

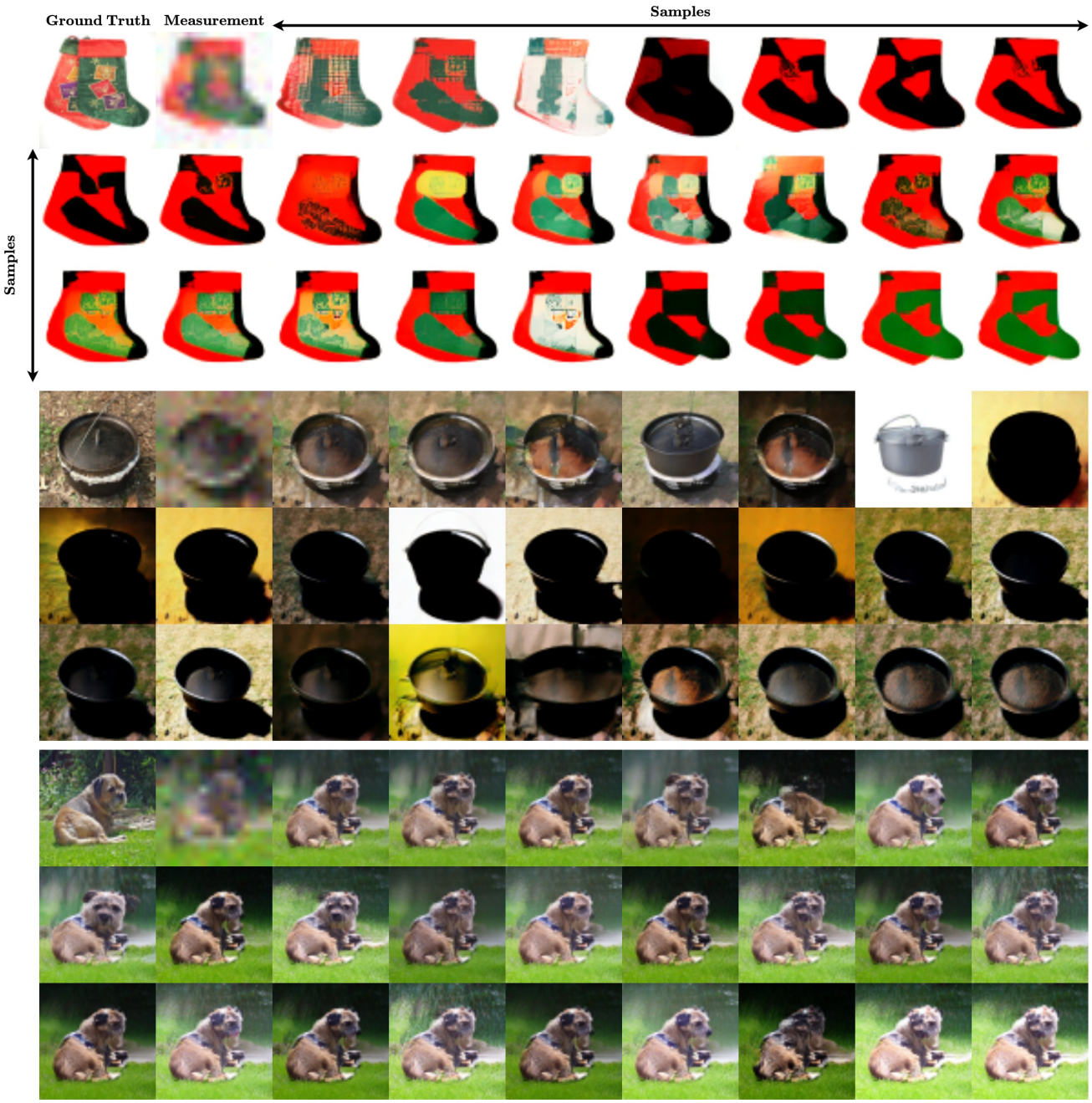

Figure A.14. Sets of samples for 4x super-resolution on ImageNet (64 x 64).

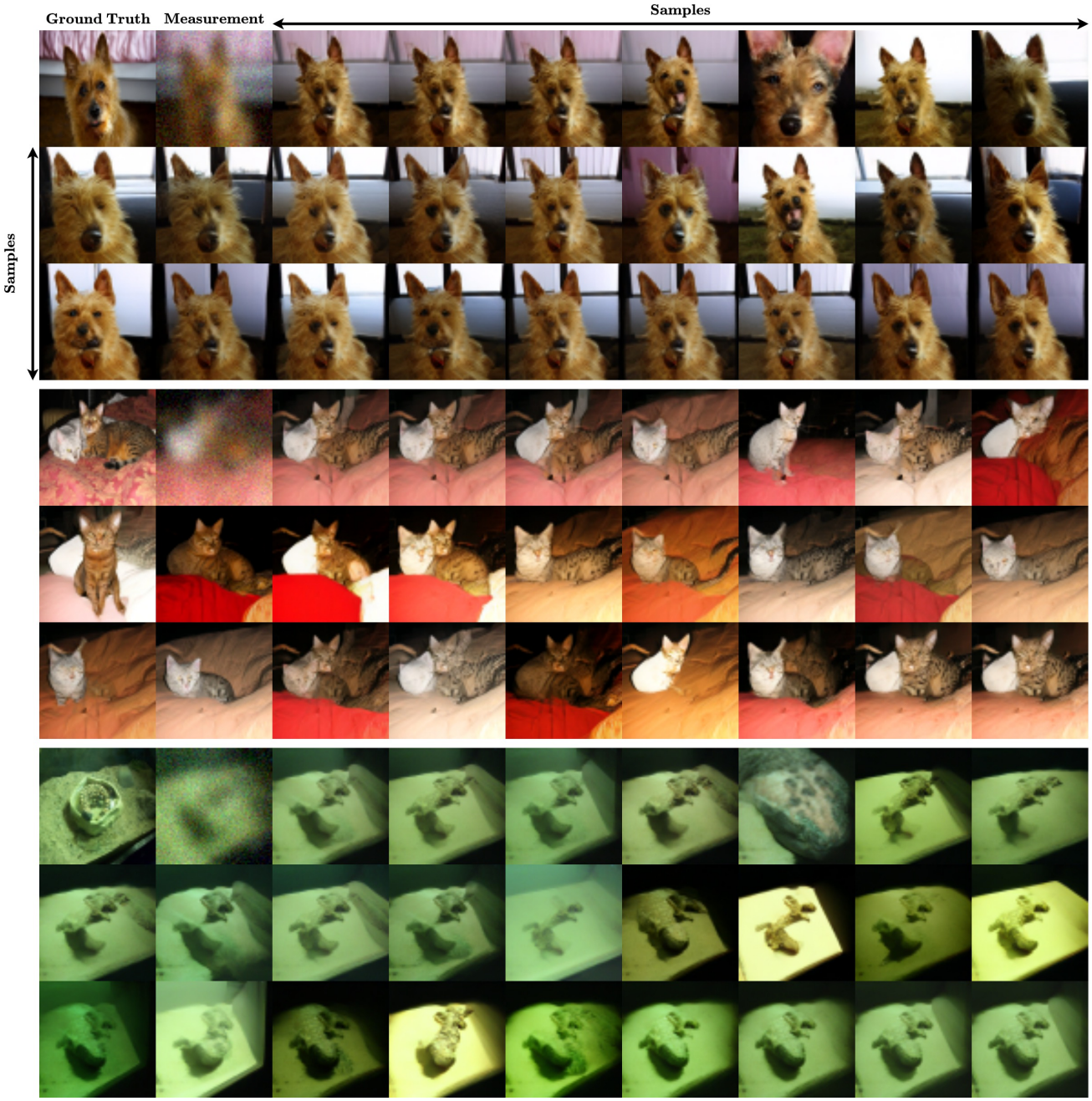

Figure A.15. Sets of samples for Gaussian deblurring on ImageNet (64 x 64).