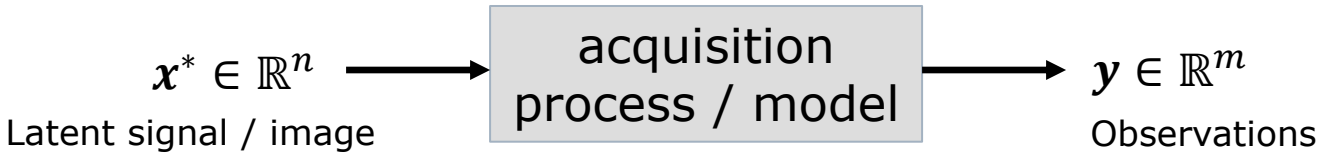
Zero-Shot Image Restoration via Few-Step Guidance of Consistency Models (and Beyond)

Tomer Garber and Tom Tirer
Bar-Ilan University, Israel

(Imaging) inverse problems



- The goal: reconstruct x^* from y
- In many image restoration tasks, the observations can be accurately modeled as

$$y = Ax^* + e$$


Ground Truth


Denoising
($A = I_n$)


Deblurring
(A is blurring)

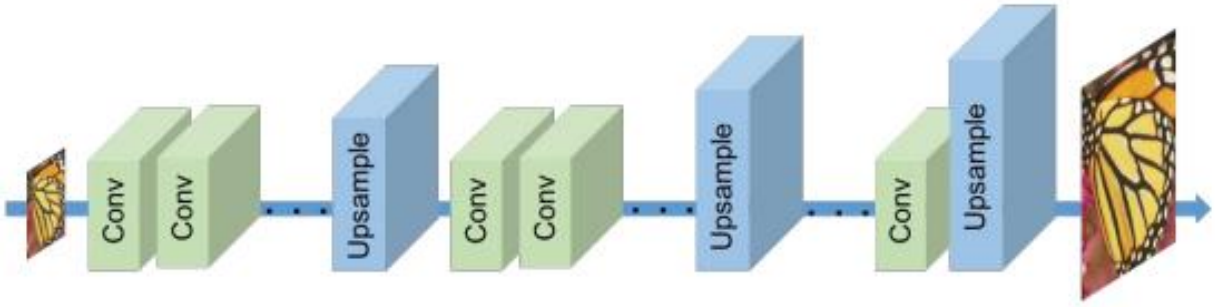

Super-Resolution
(A is blurring + downsampling)

- In all of them: just finding x that fits y is not sufficient! ("ill-posed problems")

Deep learning for inverse problems

- Most DL approaches for inverse problems:
  - Collect/synthesize a training set $\{\boldsymbol{x}_i^*, \boldsymbol{y}_i\}$ with a **predetermined** observation model
  - Learn a DNN by

$$\min_{\boldsymbol{\theta}} \sum_i \|\mathrm{DNN}_{\boldsymbol{\theta}}(\boldsymbol{y}_i) - \boldsymbol{x}_i^*\|$$

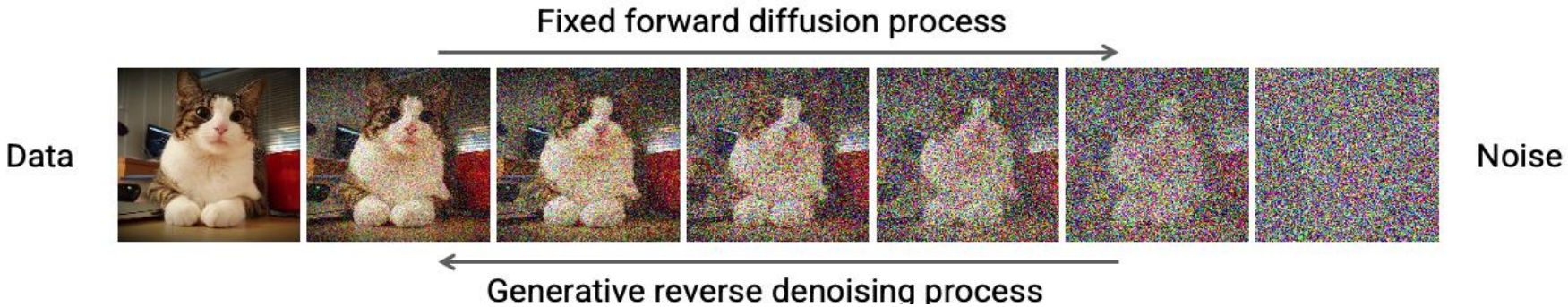


- Huge performance drop when the signal or observation model in test-time (may be unknown in advance) mismatch the training assumptions

Diffusion Models

- **Training:** Learn a model $f_\theta(x, \sigma)$ that predicts the noise
- **Sampling:** Starts from Gaussian noise, iteratively denoise and inject less noise
- Requires **dozens of NFEs** for high-quality samples.
- A typical sampling scheme with $n = N, \ldots, 1$:

$$x_{0|\tau_n} = f_\theta\left(x_{\tau_n}, \tau_n\right)$$
$$z \sim \mathcal{N}(0, \boldsymbol{I})$$
$$x_{\tau_{n-1}} = x_{0|\tau_n} + \tau_{n-1} z$$



Denoising Diffusion Implicit Models (DDIM) [Song et al., ICLR '21]

- Reduce sampling scheme NFEs by replacing the noise injection:

$$x_{\tau_{n-1}} = x_{0|\tau_n} + \tau_{n-1} z$$

$$\downarrow$$

$$x_{\tau_{n-1}} = x_{0|\tau_n} + \sqrt{1-\eta^2}\tau_{n-1}\hat{z} + \eta\tau_{n-1} z$$

- $\eta \in [0,1]$ trades between the stochastic noise and the estimated noise:

$$\hat{z} = \frac{\left(x_{\tau_n} - f_\theta\left(x_{\tau_n}, \tau_n\right)\right)}{\tau_n}$$

- High-quality image generation still requires at least three dozen NFEs

Restoration via guidance of DMs

- Pretrained DMs as prior
- Data-fidelity guidance is needed to produce relevant image
- This guidance is typically based on the gradient of a data-fidelity term $\ell(x; y)$
- In other words, the update is modified into:

$$x_{\tau_{n-1}} = x_{0|\tau_n} - \mu \nabla_x \ell\left(x_{0|\tau_n}; y\right) + \tau_{n-1} z$$

- This update is oftentimes used with the DDIM noise injection

Figure 18. CelebA-HQ: Deblurring for motion blur with noise level 0.1. [Garber & Tirer, CVPR '24]

Data-fidelity guidance

- Least squares (LS) based guidance

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_{0|t} - \mu_t \boldsymbol{A}^{\top}\left(\boldsymbol{A}\boldsymbol{x}_{0|t} - \boldsymbol{y}\right)$$ + noise injection

  follows from $\nabla_{\boldsymbol{x}} \ell_{LS}$ of $\ell_{LS}(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$

- Commonly used in PnP (general purpose denoisers as priors) and now also with DMs (e.g., "diffusion posterior sampling" (DPS) [Chung et al., ICLR '23])
- Oftentimes requires **many iterations** (many NFEs)

Data-fidelity guidance

- Back-projection (BP) based guidance

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_{0|t} - \mu_t \boldsymbol{A}^{\dagger}\left(\boldsymbol{A}\boldsymbol{x}_{0|t} - \boldsymbol{y}\right)$$ + noise injection

  where $\boldsymbol{A}_{\eta}^{\dagger} = \boldsymbol{A}^{\top}(\boldsymbol{A}\boldsymbol{A}^{\top} + \eta \boldsymbol{I}_m)^{-1}$ is the regularized pseudoinverse of $\boldsymbol{A}$.
- Oftentimes with $\mu_t = 1$.
- Proposed in IDBP [Tirer & Giryes, TIP '18] and rediscovered for DDMs ("denoising diffusion null-space" (DDNM) [Song et al., ICLR '23], "Pseudoinverse guidance" [Wang et al., ICLR '23])
- Oftentimes (much) fewer iterations than LS
- Better results at low noise (in $\boldsymbol{y}$)
- Oftentimes can be computed efficiently
- See [Garber & Tirer, CVPR '24] for further discussion and generalization

Existing methods – no less than 20 NFEs

- Current Zero-Shot image restoration methods (using DMs) require at least 20 NFEs

	Method	NFEs
[Kawar et al., NeurIPS ’22]	DDRM	20
[Zhu et al., CVPR ’23]	DiffPIR	20
[Song et al., ICLR ’23]	DDNM	100
[Garber & Tirer, CVPR ’24]	DDPG	100
[Chung et al., ICLR ’23]	DPS	1000

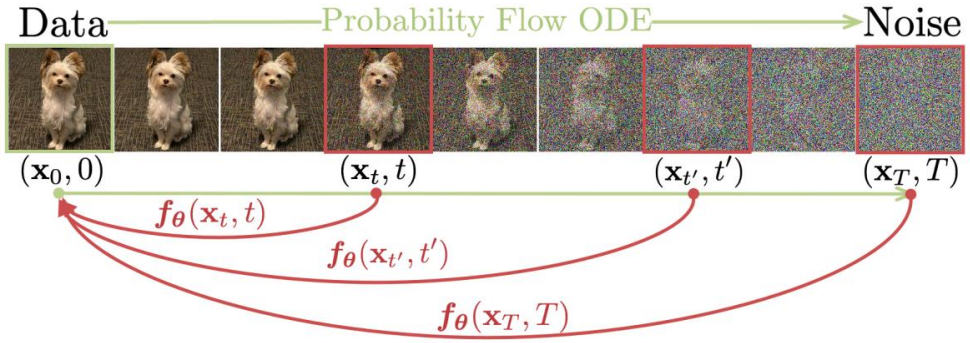
Consistency Models [Song et al., ICML '23]

- CMs improve DMs ability to map pure/very noisy samples to clean ones in a single step
- **Training:** Learn a mapping $f_\theta(x_t, t)$ so that outputs remain consistent across different time steps, by minimizing the loss:

$$\mathbb{E}\left[\lambda(t) d\left(f_\theta\left(x_{t_{n+1}}, t_{n+1}\right), f_{\theta^-}\left(\hat{x}^{\Phi}_{t_n}, t_n\right)\right)\right]$$

Distilled from pretrained DM Φ

- **Sampling:**
  - **1-step sampling**: $x_0 = f_\theta(x_T, T)$
  - **Iterative refinement**: Similar to DM sampling but with significantly fewer steps



[Song et al., ICML '23]

CM-Based image restoration

- CMs can generate high quality images with few NFEs
- However, when it comes to image restoration, existing methods requires many NFEs or task-specific tuning
- Examples include:
  - CM (40) – As part of the original CM paper, a 40 NFEs image restoration scheme was suggested
  - CoSIGN – A **task-specific** method that requires 2 NFEs

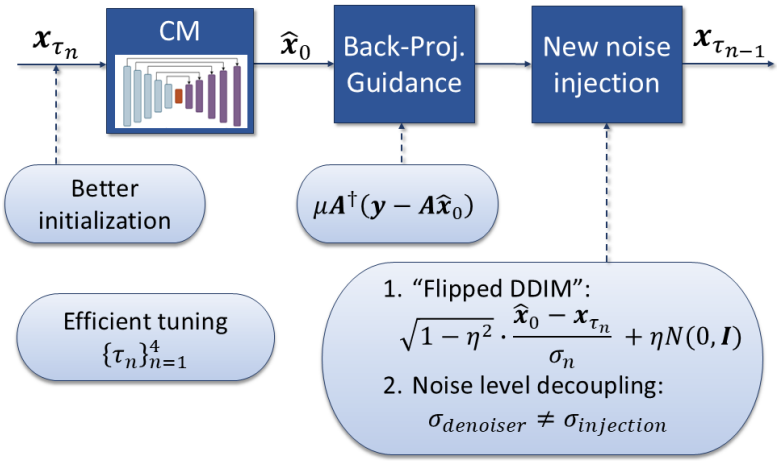
[Zhao et al., ECCV '24]

Our contributions

Zero-Shot image restoration with just **4 NFEs!**

Our approach is based on a wise combination of several ingredients:

1. Better initialization
2. Back-projection guidance
3. Novel noise injection mechanism



Better initialization

- Most of the DM based restoration techniques initialize x_{τ_N} with pure noise $x_{\tau_N} \sim \mathcal{N}(0, T^2\boldsymbol{I})$, or equivalently:

$$x_{\tau_N} = x_{init} + \tau_N z,$$

where:

$$z \sim \mathcal{N}(0, \boldsymbol{I}), x_{init} = 0 \; and \; \tau_N = T$$

- This ignores the fact that the observations vector y contains information on the specific x^* that we wish to restore.

Better initialization

- We propose to use the pseudo-inverse of A:

$$x_{init} = A^{\dagger} y$$
$$x_{\tau_N} = x_{init} + \tau_N z$$

- Examples:
  - Bicubic downsampling: $A^{\dagger}$ is bicubic upsampling
  - Deblurring: $A^{\dagger}$ is naive regularized inversion (e.g., via FFT)
  - Inpainting: $A^{\dagger} = A^{T}$. Thus, instead, we propose to use median inpainting [Tirer & Giryes, '18]

- Our initialization follows common pre-DM methods but retains noise at level τ_N in x_{init}

Noise level decoupling

- We claim that in restoration tasks: $\sigma_{denoiser} \neq \sigma_{injection}$
- This is due to:
  1. If $\sigma_y > 0$, then the guidance $\nabla_x \ell\left(x_{0|\tau_n}; y\right)$ adds noise from y into $x_{\tau_{n-1}}$
  2. Early iterations may deviate from x^*; Increasing the denoiser's noise level beyond the injection noise allows it more flexibility

- For noise injection of level τ_n we apply the CM f_θ with noise level $(1+\delta)\tau_n$ where $\delta \geq 0$ is a hyperparameter

Momentum-Like noise

- Standard CM denoising aims to move x_{τ_n} toward x_0, and to accelerate we aim to push it further in that direction
- Since x_0 is unknown, we rely on the denoiser's output $x_{0|\tau_n}$
- This leads us to define our **negative noise estimate**:

$$\hat{z}^- := \frac{\mathrm{x}_{0|\tau_n} - \mathrm{x}_{\tau_n}}{\tau_n}$$

- Noise injection become:

$$\sqrt{1-\eta^2}\tau_{n-1}\hat{z}^- + \eta\tau_{n-1}z$$

- $\eta \in [0,1]$ adjusts the trade-off between stochastic noise and our negative noise estimate
- In the paper we show Gaussian marginals property and interpretation as "noisy" Polyak's momentum

CM4IR

Require: $f_\theta(\cdot, t)$ (CM denoiser), N, $\{\tau_n\}$, $\{\mu_n\}$, δ, η, $\mathbf{A}$, $\mathbf{y}$.

1: Initialize $\mathbf{x}_{\tau_N} \sim \mathcal{N}(\mathbf{A}^\dagger \mathbf{y}, \tau_N^2 \mathbf{I}_n)$

2: **for** n from N to 1 **do**

3: $\quad \mathbf{x}_{0|\tau_n} = f_\theta(\mathbf{x}_{\tau_n}, (1+\delta)\tau_n)$

4: $\quad \mathbf{g}_{\text{BP}} = \mathbf{A}^\dagger(\mathbf{A}\mathbf{x}_{0|t} - \mathbf{y})$

5: $\quad \hat{\mathbf{z}}^- = (\mathbf{x}_{0|\tau_n} - \mathbf{x}_{\tau_n})/\tau_n$

6: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$

7: $\quad \mathbf{x}_{\tau_{n-1}} = \mathbf{x}_{0|\tau_n} - \mu_n \mathbf{g}_{\text{BP}} + \sqrt{1-\eta^2}\tau_{n-1}\hat{\mathbf{z}}^- + \eta\tau_{n-1}\mathbf{z}$

8: **end for**

9: **return** $\mathbf{x}_{0|\tau_1}$

In our experiments:
$\mu_n = 1$, $\eta = 0.1$

We use it with **N=4** NFEs!

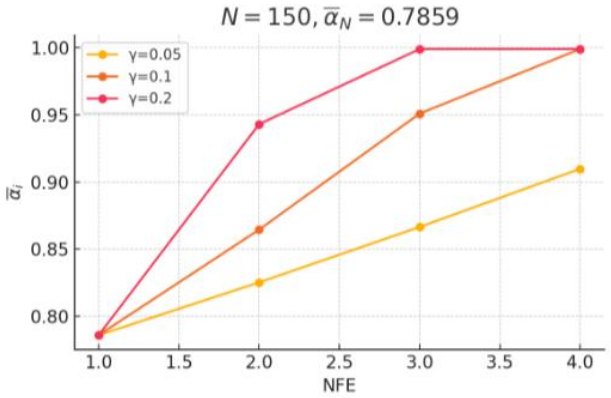
Tuning $\tau_1, \ldots, \tau_4$

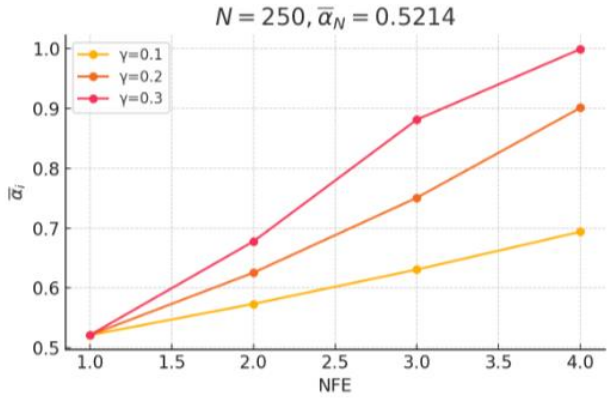
- We utilize the DDPM $\{\hat{\alpha}_i\}$ sequence
- By choosing i_N, we set $\bar{\alpha}_N = \hat{\alpha}_{i_N}$ and together with γ we define a sequence for N NFEs:

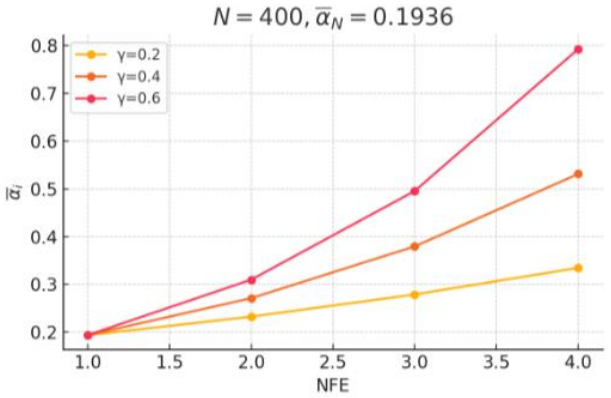
$$\bar{\alpha}_{n-1} = \bar{\alpha}_n(1+\gamma)$$

- We keep the sequence in $[0, 0.999]$
- In our experiments, we use $N = 4$ and set:

$$\tau_4 = \sqrt{1-\bar{\alpha}_{i_4}}, \ldots, \tau_1 = \sqrt{1-\bar{\alpha}_{i_1}}$$







Ablation study

Require: $f_\theta(\cdot, t)$ (CM denoiser), N, $\{\tau_n\}$, $\{\mu_n\}$, δ, η, $\mathbf{A}$, $\mathbf{y}$.

1: Initialize $\mathbf{x}_{\tau_N} \sim \mathcal{N}(\mathbf{A}^\dagger \mathbf{y}, \tau_N^2 \mathbf{I}_n)$
2: **for** n from N to 1 **do**
3: $\quad \mathbf{x}_{0|\tau_n} = f_\theta(\mathbf{x}_{\tau_n}, (1+\delta)\tau_n)$
4: $\quad \mathbf{g}_{\text{BP}} = \mathbf{A}^\dagger(\mathbf{A}\mathbf{x}_{0|t} - \mathbf{y})$
5: $\quad \hat{\mathbf{z}}^- = (\mathbf{x}_{0|\tau_n} - \mathbf{x}_{\tau_n})/\tau_n$
6: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
7: $\quad \mathbf{x}_{\tau_{n-1}} = \mathbf{x}_{0|\tau_n} - \mu_n \mathbf{g}_{\text{BP}} + \sqrt{1-\eta^2}\tau_{n-1}\hat{\mathbf{z}}^- + \eta\tau_{n-1}\mathbf{z}$
8: **end for**
9: **return** $\mathbf{x}_{0|\tau_1}$

Table 1. Ablation study on super-resolution with 4 NFEs. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Bedroom validation set.

Task \ Method	Alg.1 with δ=0 and η=1	Alg.1 with δ=0 and $\hat{\mathbf{z}}$ i/o $\hat{\mathbf{z}}^-$	Alg.1 with δ=0 and $\beta\mathbf{v}$ (Polyak) i/o $\hat{\mathbf{z}}^-$	Alg.1 with δ=0	CM4IR
SRx4 σ_y=0.025	24.49 / 0.349	24.64 / 0.348	23.37 / 0.367	25.94 / 0.298	**26.14** / 0.295
SRx4 σ_y=0.05	23.37 / 0.361	20.95 / 0.606	22.30 / 0.434	25.51 / **0.320**	**25.60 / 0.320**

Experiments

Table 2. Super-resolution, deblurring and inpainting. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Bedroom validation set.

Task \ Method	CM (40 NFEs)	CoSIGN (task spec.)	DDRM (20 NFEs)	DiffPIR (20 NFEs)	CM4IR (Ours, 4 NFEs)
SRx4 σ_y=0.025	24.66 / 0.344	26.10 / **0.205**	25.67 / 0.316	25.09 / 0.374	**26.14** / 0.295
SRx4 σ_y=0.05	23.62 / 0.449	20.35 / 0.569	25.08 / 0.354	23.83 / 0.457	**25.60 / 0.320**
Gauss. Deblurring σ_y=0.025	26.07 / 0.339	19.74 / 0.342	**28.94** / 0.221	27.48 / 0.319	28.85 / **0.217**
Gauss. Deblurring σ_y=0.05	24.18 / 0.453	19.08 / 0.543	27.35 / 0.280	26.14 / 0.363	**27.37 / 0.270**
Inpaint. (80%) σ_y=0	22.39 / 0.366	23.16 / 0.397	19.40 / 0.545	22.78 / 0.464	**25.43 / 0.284**
Inpaint. (80%) σ_y=0.025	22.17 / 0.417	23.22 / 0.368	19.16 / 0.548	22.65 / 0.477	**25.34 / 0.295**
Inpaint. (80%) σ_y=0.05	21.56 / 0.476	23.22 / 0.442	19.09 / 0.560	22.38 / 0.496	**25.28 / 0.328**

Table 3. Super-resolution, deblurring and inpainting. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Cat validation set.

Task \ Method	CM (40 NFEs)	CoSIGN (task spec.)	DDRM (20 NFEs)	DiffPIR (20 NFEs)	CM4IR (Ours, 4 NFEs)
SRx4 σ_y=0.025	25.63 / 0.366	N/A	26.93 / 0.329	26.70 / 0.349	**27.18 / 0.328**
SRx4 σ_y=0.05	24.03 / 0.459	N/A	26.05 / 0.371	25.45 / 0.399	**26.53 / 0.349**
Gauss. Deblurring σ_y=0.025	26.69 / 0.346	N/A	**29.84** / 0.258	27.93 / 0.330	29.62 / **0.246**
Gauss. Deblurring σ_y=0.05	24.54 / 0.453	N/A	**28.33** / 0.316	26.64 / 0.370	27.76 / **0.295**
Inpaint. (80%) σ_y=0.025	21.89 / 0.478	N/A	18.51 / 0.648	22.78 / 0.498	**25.89 / 0.364**
Inpaint. (80%) σ_y=0.05	21.07 / 0.523	N/A	18.48 / 0.649	22.49 / 0.514	**25.34 / 0.423**

Visual results

Figure 2. Deblurring with Gaussian kernel and noise level of 0.025. From left to right and top to bottom: original, observation, DPS [6] (1000 NFEs), DiffPIR [39] (20 NFEs), DDRM [15] (20 NFEs) and our CM4IR (4 NFEs).

Visual results

Figure 4. Inpainting (80% missing pixels) with noise level 0.05. From left to right: original, observation, DiffPIR (20 NFEs), CM (40 NFEs), CoSIGN (task specific) and our CM4IR (4 NFEs).

Visual results

Figure 1. Super-resolution ×4 with bicubic kernel and noise level of 0.05. From left to right and top to bottom: original, observation, DPS [6] (1000 NFEs), DiffPIR [39] (20 NFEs), DDRM [15] (20 NFEs) and our CM4IR (4 NFEs).

Visual results

Figure 12. Gaussian deblurring with noise level 0.05

Noise injection technique – Beyond CMs

- A key component of CM4IR is flipping the estimated noise sign compared to DDIM
- Applying this to guided DDIM-based methods (e.g., DDRM, DiffPIR) can reduce performance drops when using fewer NFEs

Table 4. Reducing NFEs for DM-based methods. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Bedroom validation set.

	Method \ NFEs, $\{\tau_n\}$	20 NFEs	4 NFEs, auto-calculated	4 NFEs, optimized	4 NFEs with our $\hat{\mathbf{z}}^-$ instead of $\hat{\mathbf{z}}$
SRx4, $\sigma_y = 0.025$	DDRM	25.67 / **0.316**	24.16 / 0.395	25.40 / 0.325	**25.89** / 0.327
SRx4, $\sigma_y = 0.025$	DiffPIR	25.09 / 0.374	24.52 / 0.450	24.68 / 0.425	**25.51** / **0.371**
SRx4, $\sigma_y = 0.05$	DDRM	25.08 / **0.354**	24.12 / 0.396	24.85 / 0.361	**25.22** / 0.364
SRx4, $\sigma_y = 0.05$	DiffPIR	23.83 / 0.457	23.32 / 0.519	23.42 / 0.506	**24.89** / **0.404**

Noise injection technique – Beyond CMs

Figure 6. Super-resolution with noise level 0.025. From left to right: original, observation, DDRM(20 NFEs), DDRM(4 NFEs, auto-calculated), DDRM(4 NFEs, optimized) and DDRM(4 NFEs with our $\hat{\mathbf{z}}^-$ instead of $\hat{\mathbf{z}}$).

(a) Original (b) Observation (c) 20 NFEs (d) 4 NFEs, auto (e) 4 NFEs, optimized (f) 4 NFEs, ours

Figure 20. Reducing NFEs for DiffPIR, Super-resolution with $\sigma_y = 0.025$

Thank you!

Many experiments and analyses can be found in:

Garber, T. and Tirer, T., "Zero-Shot Image Restoration Using Few-Step Guidance of Consistency Models (and Beyond)," Accepted to CVPR 2025

https://github.com/tirer-lab/CM4IR