A. Proof of Proposition 3.1

Proposition. *For $\eta \in [0,1]$, consider the distribution:*

$$q_\eta(\mathbf{x}_{\tau_1:\tau_N}|\mathbf{x}_0) = q_\eta(\mathbf{x}_{\tau_N}|\mathbf{x}_0)\prod_{n=2}^{N} q_\eta(\mathbf{x}_{\tau_{n-1}}|\mathbf{x}_{\tau_n},\mathbf{x}_0) \tag{25}$$

where $q_\eta(\mathbf{x}_{\tau_N}|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0,\tau_N^2\mathbf{I})$, as in [22]. For all $n>1$, let

$$q_\eta(\mathbf{x}_{\tau_{n-1}}|\mathbf{x}_{\tau_n},\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_0 + \xi\sqrt{1-\eta^2}\tau_{n-1}\frac{\mathbf{x}_0-\mathbf{x}_{\tau_n}}{\tau_n}, \eta^2\tau_{n-1}^2\mathbf{I}\right). \tag{26}$$

where $\xi \in \{-1,1\}$. Then, for for any $n\in[1,N]$ we have that

$$q_\eta(\mathbf{x}_{\tau_n}|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0,\tau_n^2\mathbf{I}). \tag{27}$$

Proof. The proof follows the one of Lemma 1 in [22], which is based on induction. For the base case $n=N$, (27) holds by construction, and under the induction assumption for n, i.e., $q_\eta(\mathbf{x}_{\tau_n}|\mathbf{x}_0)=\mathcal{N}(\mathbf{x}_0,\tau_n^2\mathbf{I})$, one needs to prove for $n-1$.

As pointed in [22], the marginal $q_\eta(\mathbf{x}_{\tau_{n-1}}|\mathbf{x}_0) = \int_{\mathbf{x}_{\tau_n}} q_\eta(\mathbf{x}_{\tau_{n-1}}|\mathbf{x}_{\tau_n},\mathbf{x}_0)q_\eta(\mathbf{x}_{\tau_n}|\mathbf{x}_0)d\mathbf{x}_{\tau_n}$ is Gaussian $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ with

$$\boldsymbol{\mu} = \mathbf{x}_0 + \xi\sqrt{1-\eta^2}\tau_{n-1}\frac{\mathbf{x}_0-\mathbb{E}\{\mathbf{x}_{\tau_n}|\mathbf{x}_0\}}{\tau_n} = \mathbf{x}_0 + \xi\sqrt{1-\eta^2}\tau_{n-1}\frac{\mathbf{x}_0-\mathbf{x}_0}{\tau_n} = \mathbf{x}_0$$

$$\boldsymbol{\Sigma} = \xi^2\frac{(1-\eta^2)\tau_{n-1}^2}{\tau_n^2}\mathrm{Cov}(\mathbf{x}_{\tau_n}|\mathbf{x}_0) + \eta^2\tau_{n-1}^2\mathbf{I} = \frac{(1-\eta^2)\tau_{n-1}^2}{\tau_n^2}\tau_n^2\mathbf{I} + \eta^2\tau_{n-1}^2\mathbf{I} = \tau_{n-1}^2\mathbf{I}$$

where we used $\xi^2=1$.

□

B. More Experimental Details and Results

In this section, we provide additional details about the experiments, as well as supplementary quantitative and qualitative results that were not included in the main body of the paper due to space constraints. Our code for reproducing the results will be made available upon acceptance. Our code is available at https://github.com/tirer-lab/CM4IR.

B.1. Hyperparameter setting

In our experiments, we utilize the DDPM sequence $\{\beta_i\}_{i=1}^{1000}$ for setting the chosen time points / noise levels. Specifically, consistent with many DM methods, the sequence $\{\beta_i\}$ is defined using a linear schedule that ranges from $\beta_1 = 0.0001$ to $\beta_{1000} = 0.02$. A sequence $\{\hat{\alpha}_i\}$ is computed as follows: $\hat{\alpha}_i = \prod_{j=1}^{i}\alpha_j$ with $\alpha_j = 1-\beta_j$. Observe that $\{\hat{\alpha}_i\}$ decreases as i increases. Given an index number i_N, we select the relevant $\hat{\alpha}_{i_N}$ from $\{\hat{\alpha}_i\}$ and set $\overline{\alpha}_N = \hat{\alpha}_{i_N}$.

As explained in Section 4, in CM4IR we use γ and $\overline{\alpha}_N$ to tune the sequence $\{\overline{\alpha}_n\}_{n=1}^{N}$ for the N NFEs, in the following manner: $\overline{\alpha}_{n-1} = \overline{\alpha}_n(1+\gamma)$. We keep the sequence bounded within the range $[0, 0.999]$.

In Figure 7, we observe sequences $\{\overline{\alpha}_n\}$ of length 4 for various i_N values and corresponding γ settings. The plots illustrate how the choice of γ influences the progression of $\overline{\alpha}_n$ values over time steps, starting from an initial $\overline{\alpha}_N$. The left plot corresponds to $i_N = 150$ with smaller γ values, the middle plot represents $i_N = 250$ with moderate γ values, and the right plot shows $i_N = 400$ with higher γ settings.

The hyperparameters of CM4IR are listed in Table 5. The tuning shows a high level of consistency, making our method intuitive and easy to tune across tasks and datasets. For instance, η remains fixed at 0.1 in all configurations. Similarly, γ and i_N values exhibit only minor variations within each task, showing that fine-tuning requires minimal effort. For the deblurring task, we also regularize the BP in the intermediate iteration. Namely, we use $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \sigma_y^2\zeta)^{-1}$ with $\zeta > 0$.

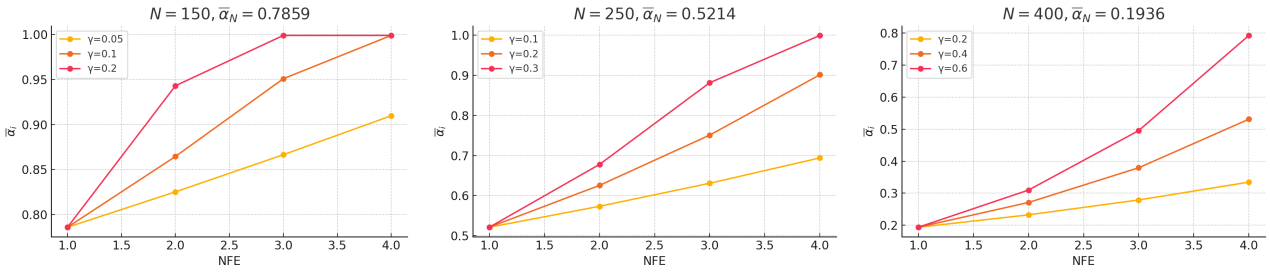

Figure 7. $\overline{\alpha}_n$ sequences for different i_N and γ settings. $\overline{\alpha}_n$ values are clipped to $[0, 0.999]$. Recall that the noise level is $\tau_n = \sqrt{1 - \overline{\alpha}_n}$.

Table 5. CM4IR hyperparameters.

Task	LSUN Bedroom	LSUN Cat
Bicub. SRx4 σ_e=0.025	$\eta = 0.1, i_N = 400, \gamma = 0.7, \delta_i = (0.0, 0.5, 0.1, 0.0)$	$\eta = 0.1, i_N = 400, \gamma = 0.7, \delta_i = (0.0, 0.3, 0.0, 0.0)$
Bicub. SRx4 σ_e=0.05	$\eta = 0.1, i_N = 250, \gamma = 0.2, \delta_i = (0.0, 0.3, 0.05, 0.1)$	$\eta = 0.1, i_N = 250, \gamma = 0.2, \delta_i = (0.1, 0.1, 0.0, 0.0)$
Gauss. Deblurring σ_e=0.025	$\eta = 0.1, i_N = 90, \gamma = 0.02, \zeta = 3.0, \delta_i = (0.0, 0.0, 0.0, 0.0)$	$\eta = 0.1, i_N = 100, \gamma = 0.03, \zeta = 4.0, \delta_i = (0.0, 0.0, 0.0, 0.0)$
Gauss. Deblurring σ_e=0.05	$\eta = 0.1, i_N = 160, \gamma = 0.07, \zeta = 1.5, \delta_i = (0.0, 0.0, 0.0, 0.0)$	$\eta = 0.1, i_N = 180, \gamma = 0.1, \zeta = 2.0, \delta_i = (0.0, 0.0, 0.0, 0.0)$
Inpaint. (80%) σ_e=0	$\eta = 0.1, i_N = 150, \gamma = 0.2, \delta_i = (0.1, 0.1, 0.8, 0.8)$	N/A
Inpaint. (80%) σ_e=0.025	$\eta = 0.1, i_N = 150, \gamma = 0.2, \delta_i = (0.2, 0.3, 0.8, 0.8)$	$\eta = 0.1, i_N = 150, \gamma = 0.2, \delta_i = (0.0, 0.0, 1.0, 1.0)$
Inpaint. (80%) σ_e=0.05	$\eta = 0.1, i_N = 150, \gamma = 0.2, \delta_i = (0.2, 0.1, 1.0, 1.0)$	$\eta = 0.1, i_N = 150, \gamma = 0.2, \delta_i = (0.0, 0.0, 1.0, 1.0)$

B.2. Robustness to hyperparameter settings

To asses the robustness of CM4IR to the hyperparameter settings, we examine PSNR/LPIPS for SRx4 with σ_y=0.025 on the LSUN bedroom when we moderately modify the parameters from the values stated in B.1. Figure 8 (left) shows that increasing η has only minor effect on the PSNR/LPIPS. Consistent PSNR/LPIPS are also observed in Figure 8 (right) when decreasing (i_N, γ). Recall that i_N determines the first noise level (higher i_N implies stronger noise) and γ determines its decay rate. Thus, it makes sense to pair them, as examined.

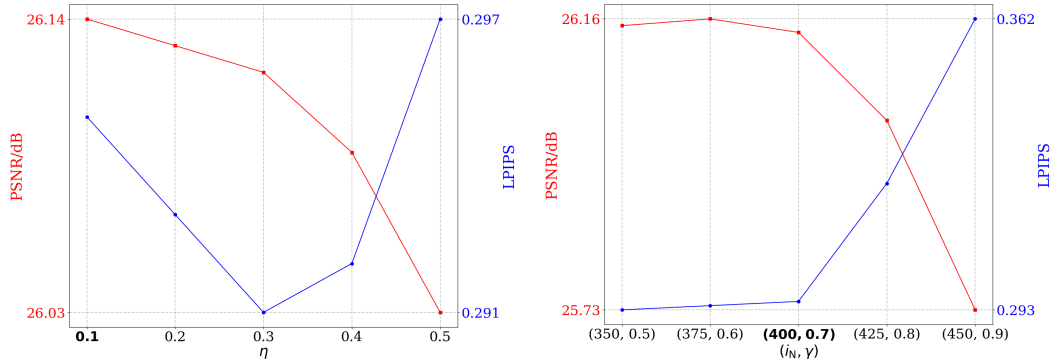

Figure 8. PSNR/LPIPS for different parameters. The chosen values are in **bold**.

B.3. Effectiveness of δ

In Table 1, we presented some of the effectiveness of using $\delta > 0$ within the ablation study. The results presented in Tables 6 and 7 demonstrate the effectiveness of using $\delta > 0$ across more tasks and datasets. In most cases, introducing non-zero δ values consistently improves the PSNR. Additionally, setting $\delta > 0$ often leads to better (or similar) LPIPS.

Table 6. Super-resolution, gaussian deblurring and inpainting. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Bedroom validation set.

Task \ δ	Alg.1 with $\delta = 0$	Alg.1 with $\delta > 0$
SRx4 σ_y=0.025	25.94 / 0.298	**26.14 / 0.295**
SRx4 σ_y=0.05	25.51 / **0.320**	**25.60 / 0.320**
Inpaint. (80%) σ_y=0	25.13 / 0.296	**25.43 / 0.284**
Inpaint. (80%) σ_y=0.025	25.05 / 0.301	**25.34 / 0.295**
Inpaint. (80%) σ_y=0.05	24.85 / 0.363	**25.28 / 0.328**

Table 7. Super-resolution and inpainting. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Cat validation set.

Task \ δ	Alg.1 with $\delta = 0$	Alg.1 with $\delta > 0$
SRx4 σ_y=0.025	27.04 / **0.325**	**27.18** / 0.328
SRx4 σ_y=0.05	26.44 / **0.342**	**26.53** / 0.349
Inpaint. (80%) σ_y=0.025	25.75 / 0.381	**25.89 / 0.364**
Inpaint. (80%) σ_y=0.05	25.13 / 0.424	**25.34 / 0.423**

B.4. Different amount of NFEs

In this subsection, we investigate alternative choices for the number of NFEs. To provide a comprehensive evaluation, we adapted the same algorithm for setups with 3 NFEs and 5 NFEs, maintaining $\delta = 0$ across all cases. Each configuration required tuning of the i_N, γ and η hyperparameters. For $N = 3$ NFEs, we tuned i_N and γ for obtaining three $\overline{\alpha}_n$ values, and for $N = 5$ NFEs we tuned i_N and γ for obtaining five $\overline{\alpha}_n$ values.

The results of these experiments are summarized in Table 8. As the table indicates, while the 3 NFEs configuration achieves reasonable performance, it does not fully utilize the reconstruction capacity of the algorithm. On the other hand, increasing to 5 NFEs does not show gains compared to 4 NFEs. This may be due to restrictions of our tuning strategy that uses only two hyperparameters.

Table 8. Evaluating CM4IR with different amount of NFEs. Super-resolution. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Bedroom validation set.

Task \ NFEs	3 NFEs	4 NFEs	5 NFEs
SRx4 σ_y=0.025	25.71 / 0.319	25.94 / **0.298**	**25.98** / 0.315
SRx4 σ_y=0.05	25.31 . 0.348	**25.51 / 0.320**	25.42 / 0.339

B.5. Advantages over additional competitors

In this subsection, we evaluate DPS [6], a reconstruction method based on pretrained DMs with Least-Squares (LS) guidance. For super-resolution, we tested DPS under the settings $\sigma_y = 0.025$ and $\sigma_y = 0.05$ on the LSUN Bedroom dataset. DPS achieved PSNR/LPIPS results of 24.81 / 0.362 and 24.08 / 0.400 for these noise levels, respectively, which are significantly lower than the performance of our proposed CM4IR method.

Additionally, DPS requires 1000 NFEs per image, coupled with a much larger per-iteration computational complexity due to computing the model's Jacobian, resulting in an extremely high computational cost. Each image takes approximately 75 seconds to process on the same hardware and using the same DM as the rest of the methods.

In contrast, our CM4IR method operates with only 4 NFEs, reducing the computational time to just 0.159 seconds per image while delivering superior results.

B.6. Reducing NFEs of additional DM-based method

DDNM [34] is another reconstruction method that leverages pretrained DMs with Back-Projection guidance [28]. As highlighted in [9], this method is designed and performs well for noiseless observations. Notice that DDNM utilizes guided DDIM scheme, so like DDRM and DiffPIR, we can explore if its performance drop when reducing the NFEs can be mitigated by our modified noise injection.

In Table 9, we present results for reducing the NFEs of DDRM, DiffPIR and DDNM in the noiseless super-resolution task

using the LSUN Bedroom validation set. As we can see, our noise injection mechanism improves PSNR, while mitigating LPIPS drops.

Table 9. 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, $\sigma_y = 0.0$	DDRM	26.27 / **0.251**	25.90 / 0.272	25.92 / 0.268	**26.31** / 0.261
SRx4, $\sigma_y = 0.0$	DiffPIR	25.92 / 0.287	25.73 / 0.309	25.69 / 0.319	**26.30** / **0.271**
SRx4, $\sigma_y = 0.0$	DDNM	26.33 / **0.242**	25.93 / 0.268	26.03 / 0.263	**26.38** / 0.262

B.7. Additional inpainting results

In this subsection, we explore additional inpainting scenario - removal of superimposed text. For reproducibility and fair comparison we use the same text mask, shown in Figure 14, for all images, and the same median initialization for all methods. The results in Table 10 show the advantages of our method.

Table 10. Removal of superimposed text. PSNR [dB] (↑) and LPIPS (↓) results on LSUN Bedroom validation set.

Task \ Method	CM (40)	CoSIGN (task spec.)	DDRM (20)	DiffPIR (20)	CM4IR (Ours)
Inpaint. (text) σ_y=0.025	30.86 / 0.189	23.90 / 0.296	33.33 / 0.119	30.95 / 0.198	**34.94** / **0.087**

B.8. Additional dataset

In this section, we evaluate CM4IR also for ImageNet 64×64 for SR with bicubic downsampling of factor 2 and different noise levels. This factor allows to obtain meaningful recoveries despite the low resolution of the original 64×64 images. The results are presented in Table 11, and show that CM4IR outperforms the other methods in terms of PSNR and has competitive LPIPS despite using only 4 NFEs.

Table 11. Bicubic SRx2 for ImageNet 64x64. PSNR [dB] and LPIPS.

Task \ Method	CM (40 NFEs)	DDRM (20 NFEs)	DiffPIR (20 NFEs)	CM4IR (Ours, 4 NFEs)
SRx2, σ_y=0.01	27.33 / 0.165	29.26 / **0.104**	28.64 / 0.132	**30.11** / 0.122
SRx2, σ_y=0.025	25.98 / 0.235	28.38 / **0.134**	27.10 / 0.163	**29.20** / 0.148
SRx2, σ_y=0.05	24.51 / 0.309	27.02 / **0.178**	23.65 / 0.250	**27.74** / 0.188

B.9. More qualitative results

In this subsection, we present visual results for the different tasks.

Figure 9. Gaussian deblurring with noise level 0.025

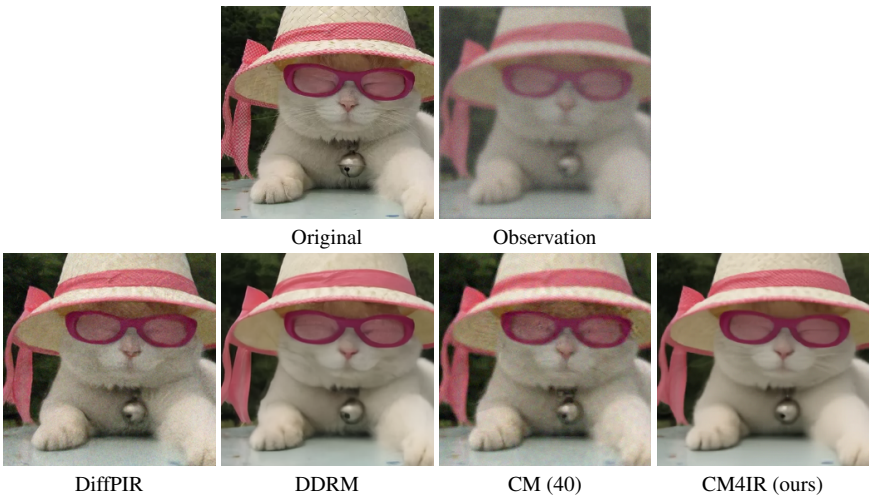

Figure 10. Gaussian deblurring with noise level 0.025

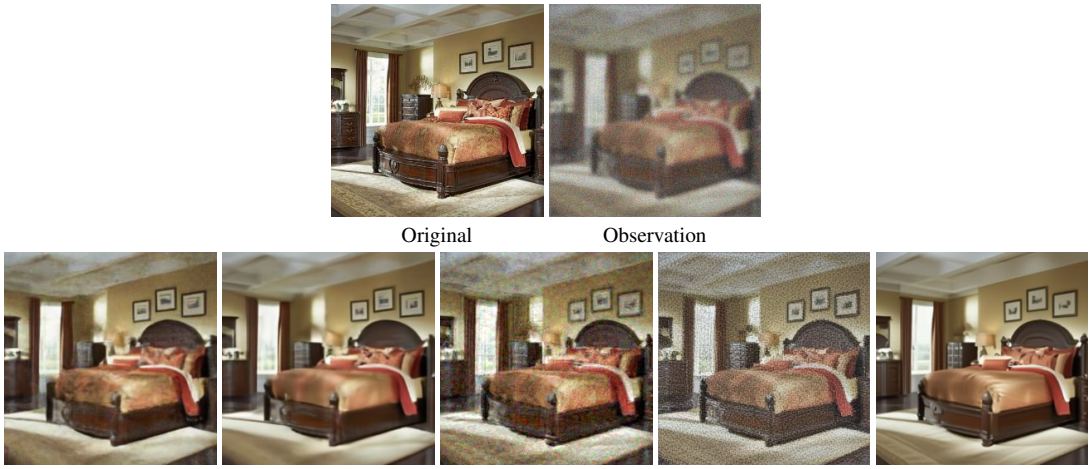

Figure 11. Gaussian deblurring with noise level 0.05

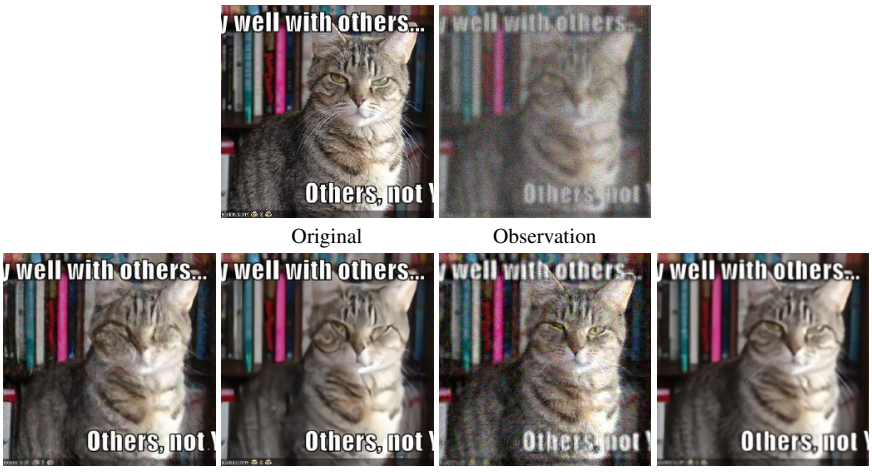

Figure 12. Gaussian deblurring with noise level 0.05

Figure 13. Removal of superimposed text with noise level 0.025

Figure 14. Removal of superimposed text with noise level 0.025

Figure 15. Super-resoulution (80%) with noise level 0.025

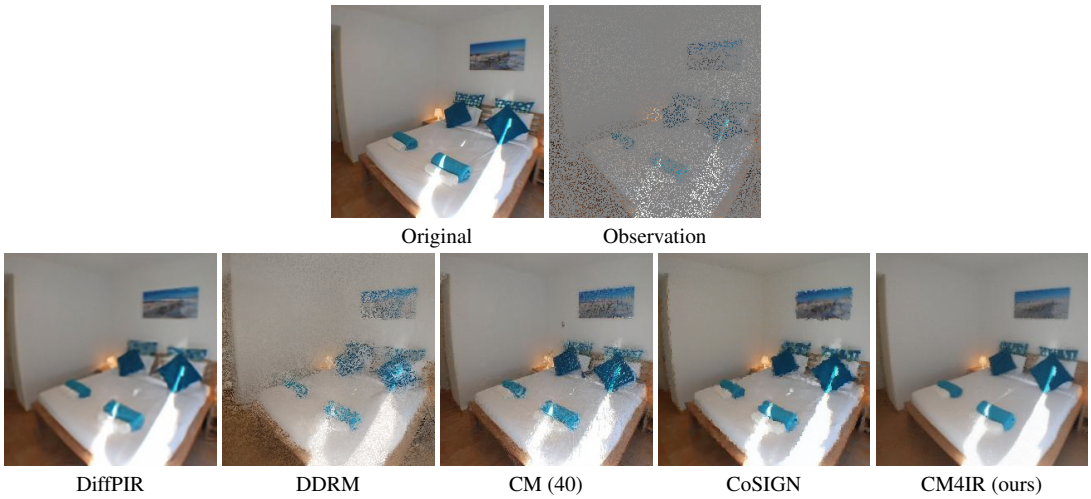

Figure 16. Inpainting (80%) with noise level 0.0

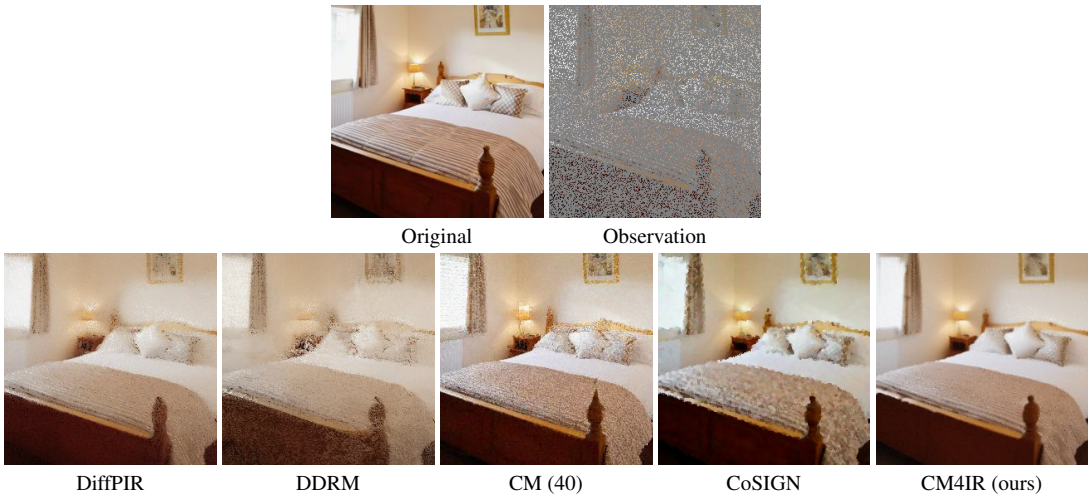

Figure 17. Inpainting (80%) with noise level 0.025

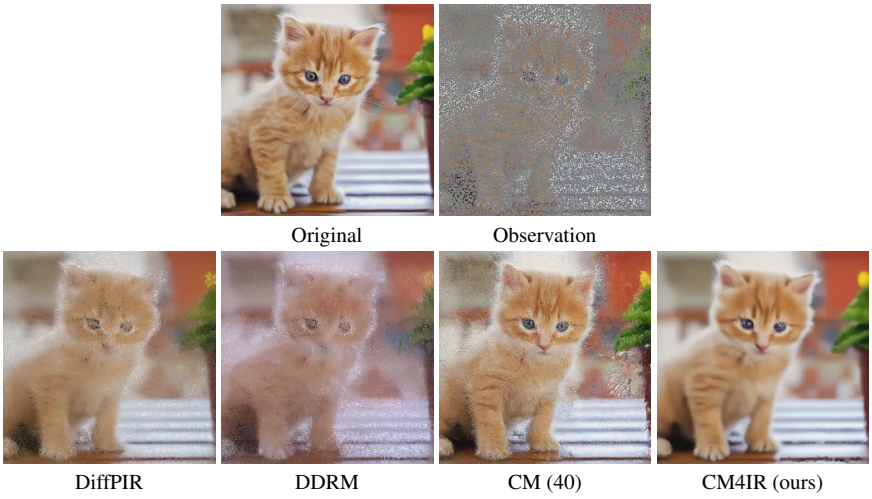

Figure 18. Inpainting (80%) with noise level 0.025

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

Figure 19. Reducing NFEs for DDNM, Super-resolution with $\sigma_y = 0.0$

(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 $\sigma_y = 0.025$

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

Figure 21. Reducing NFEs for DDRM, Super-resolution with $\sigma_y = 0.05$