Appendix for Inverse Problems with Diffusion Models: A MAP Estimation Perspective

A. Runtime Analysis

In MAP-GA (Algorithm 2), per iteration i.e. when $num\_steps = num\_iter = 1$, we have 2 NFE (Neural Function Evaluations i.e. forward pass for inference) of the consistency model, 1 NFE of the denoiser and a vector-Jacobian product computation using the consistency model. Computational cost is 2*$O(C_\theta) + O(D_\theta) + O(vjpC_\theta)$, where $O(C_\theta)$ (for a lack of better notation), denotes the cost of 1 NFE of the consistency model, $O(D_\theta)$ denotes the cost of 1 NFE of the denoiser and $O(vjpC_\theta)$ denotes the cost of 1 backward pass for the vector-Jacobian product using the consistency model.

Similarly, the computational costs per iteration for MAP-GA variants, PGDM, DDRM, and CT-ZSIE are reported in Tab 1. (We assume that in MAP-GA and variants, and PGDM, the cost for computing the terms such as $grad_{likelihood}$ involving the matrix multiplication and inverse is negligible compared to an NFE. We use efficient SVD decomposition of the forward operator matrix H for several image restoration problems to make matrix inverse computation efficient. So we assume that terms like $grad_{likelihood}$ can be computed efficiently in practice).

In our case, the pre-trained denoiser and the consistency models have an exactly similar model architecture (except in the final layer), so the computational costs for using the denoiser and the consistency model are roughly similar i.e. $O(C_\theta) \approx O(D_\theta)$, and $O(vjpC_\theta) \approx O(vjpD_\theta)$. The runtime comparison from Tab 5 in the main paper also validates the table below.

Method	Computational cost per iteration
MAP-GA	$2 * O(C_\theta) + O(D_\theta) + O(vjpC_\theta)$
MAP-GA(D)	$2 * O(D_\theta) + O(D_\theta) + O(vjpD_\theta)$
MAP-GA(NP)	$2 * O(C_\theta) + O(vjpC_\theta)$
MAP-GA(D,NP)	$2 * O(D_\theta) + O(vjpD_\theta)$
PGDM	$O(D_\theta) + O(vjpD_\theta)$
DDRM	$O(D_\theta)$
CT-ZSIE	$O(C_\theta)$

Table 1. Computational cost of MAP-GA (and variants), and other baselines. Note that MAP-GA(D) and MAP-GA(D,NP) indicate that the consistency model is replaced with the denoiser in MAP-GA and MAP-GA(NP) respectively. One iteration in PGDM, DDRM, and CT-ZSIE means one backward diffusion step.

B. Qualitative examples

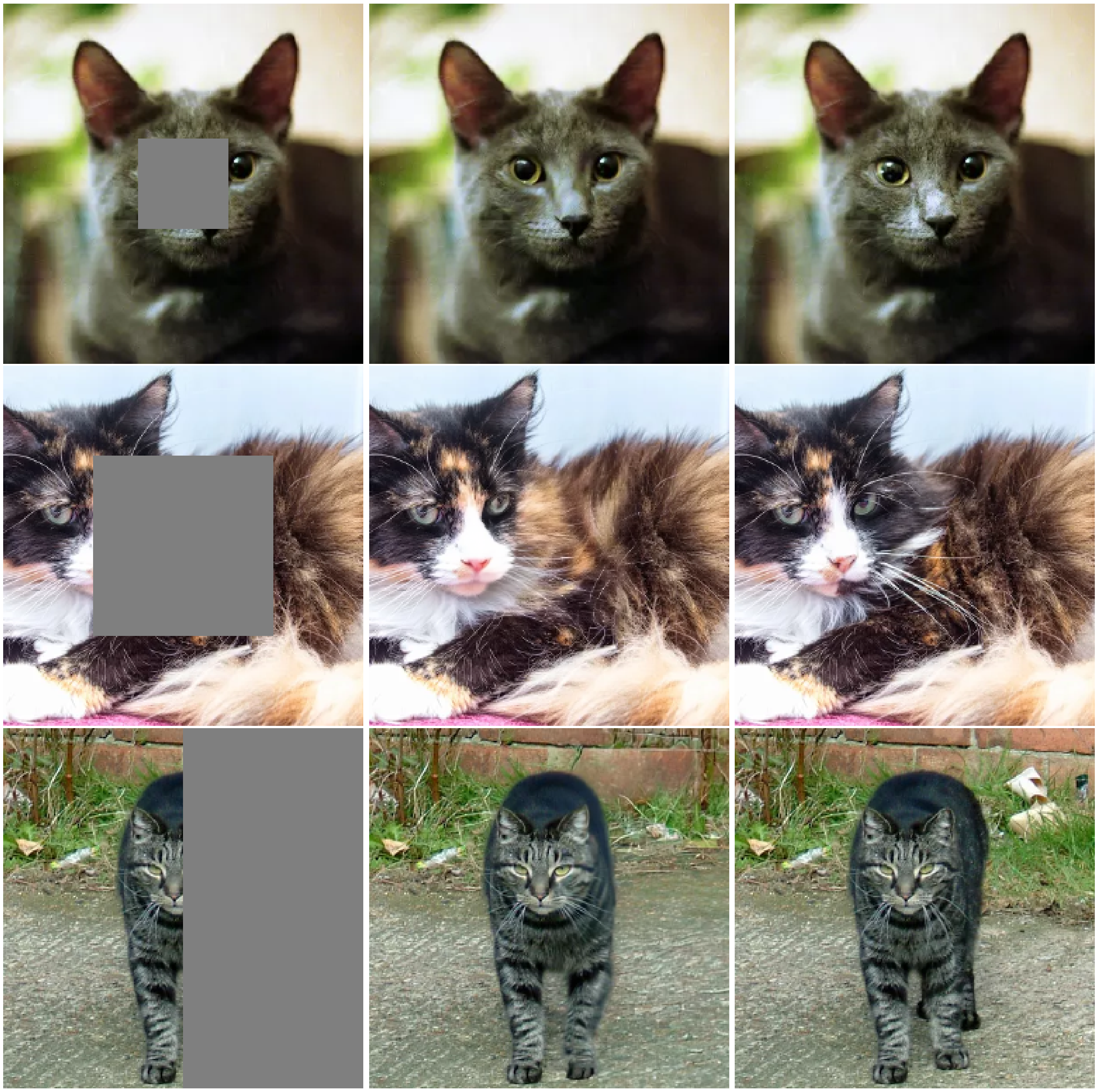

Figure 1. MAP-GA(D,NP) based image restoration on LSUNCat256. Left to right: observed, recovered, original images. Top to bottom: mask settings *box25*, *box50*, *half*. We choose $num\_steps = 20$, $num\_iter = 50$

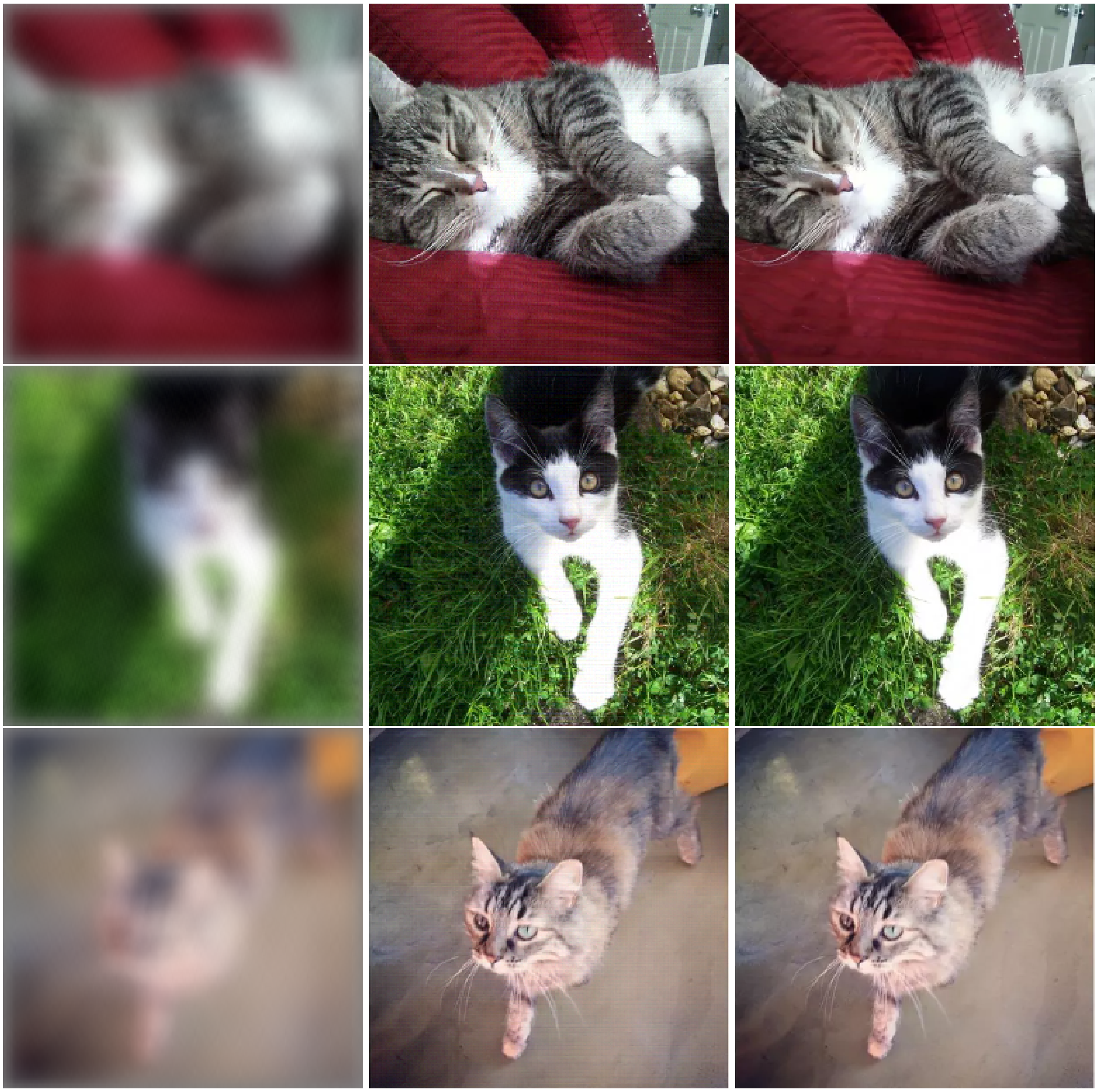

Figure 2. MAP-GA(D,NP) based image deblurring (with 16×16 uniform blur kernel) on LSUNCat256. Left to right: observed, recovered, original images. We choose $num\_steps = 20$, $num\_iter = 50$.

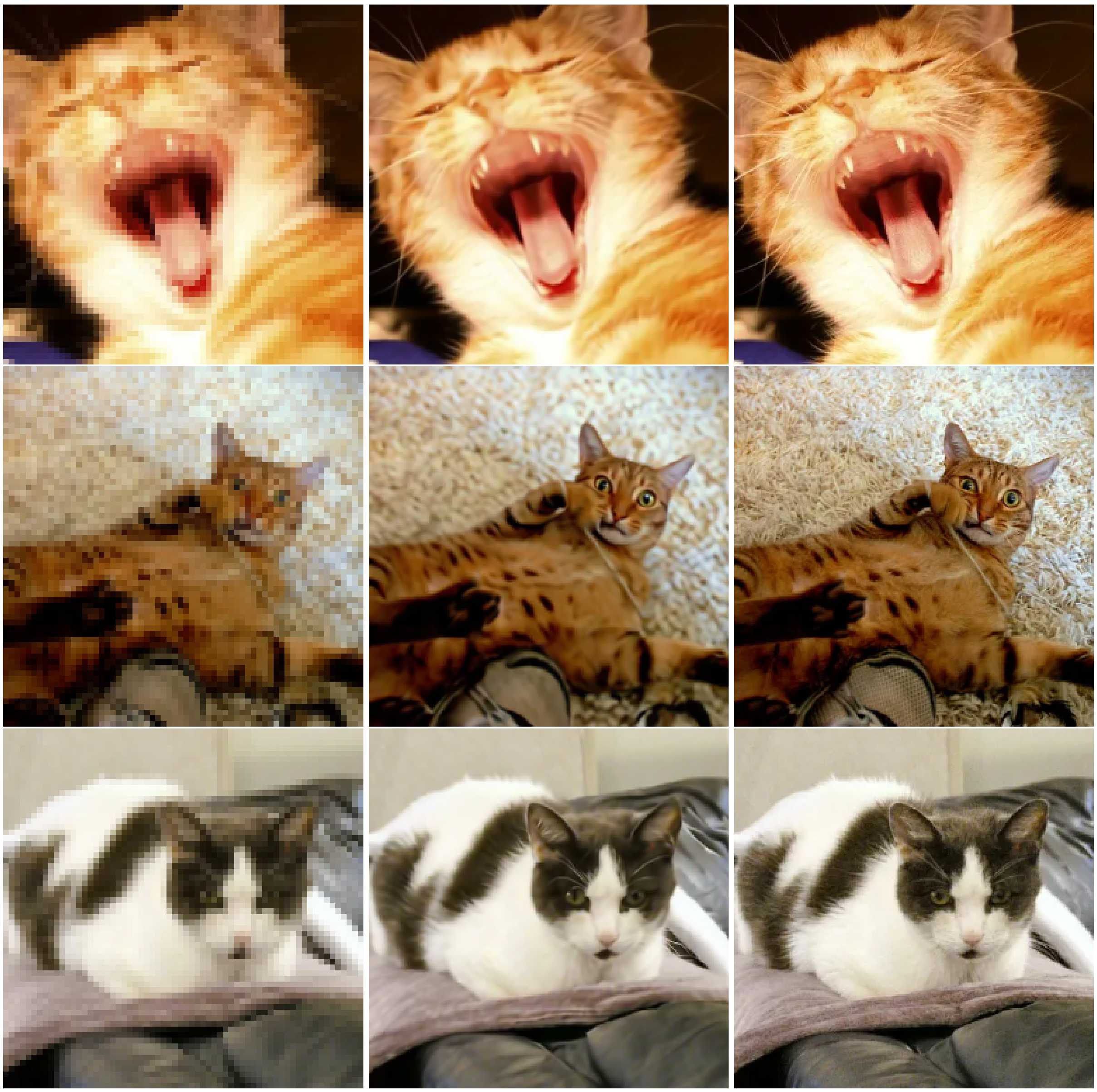

Figure 3. MAP-GA(D,NP) based $4\times$ image super-resolution on LSUNCat256. Left to right: observed, recovered, original images. We choose $num\_steps = 20$, $num\_iter = 50$.

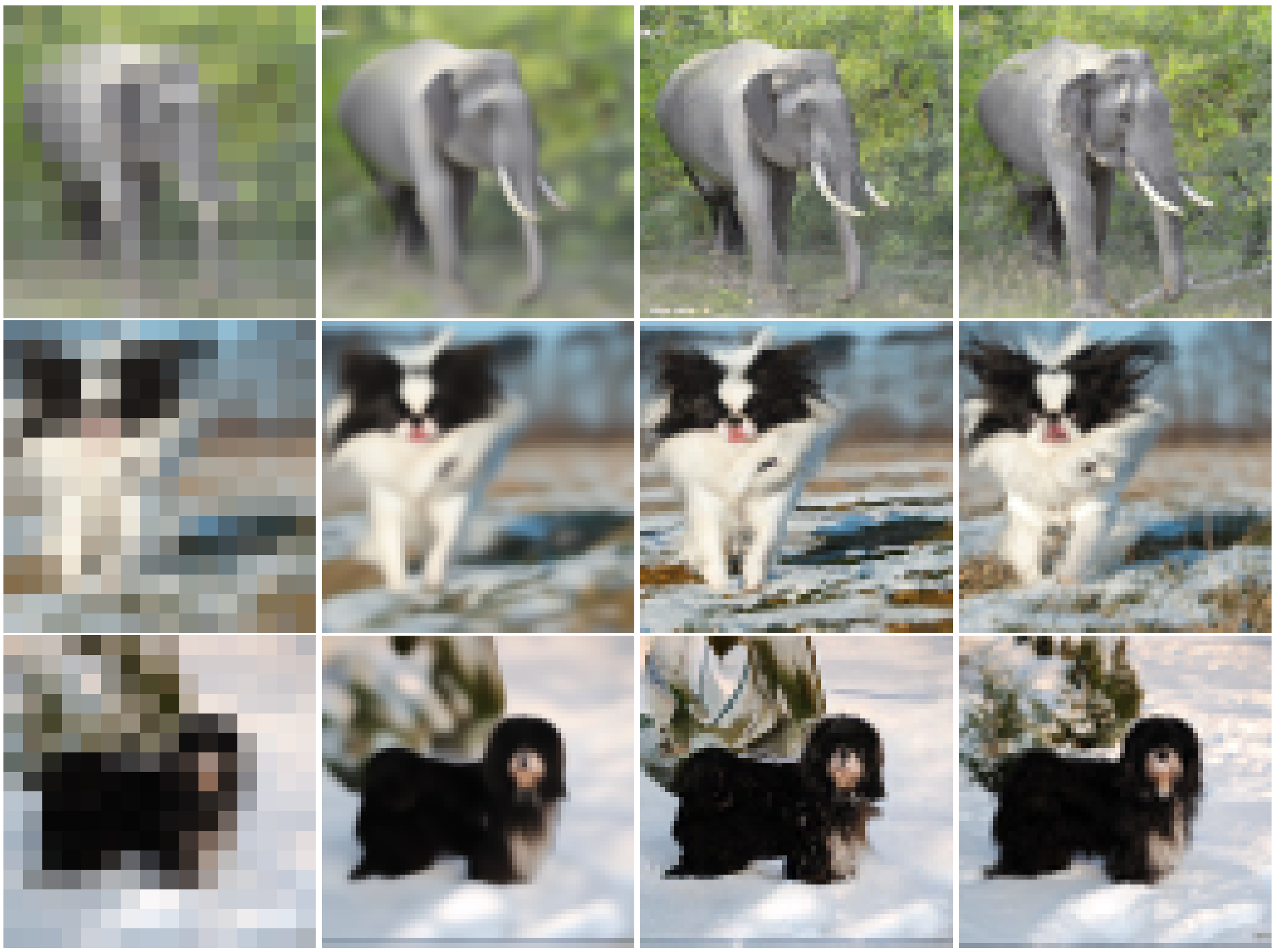

Figure 4. $4\times$ image super-resolution on ImageNet64. Left to right: observed image, recovered image using MAP-GA(D,NP), recovered image using MAP-GA(NP), original image. We choose $num\_steps = 20$, $num\_iter = 50$ for both variants. Note that this $4\times$ downsampling corresponds to a severe degradation since the original image is of 64×64 resolution

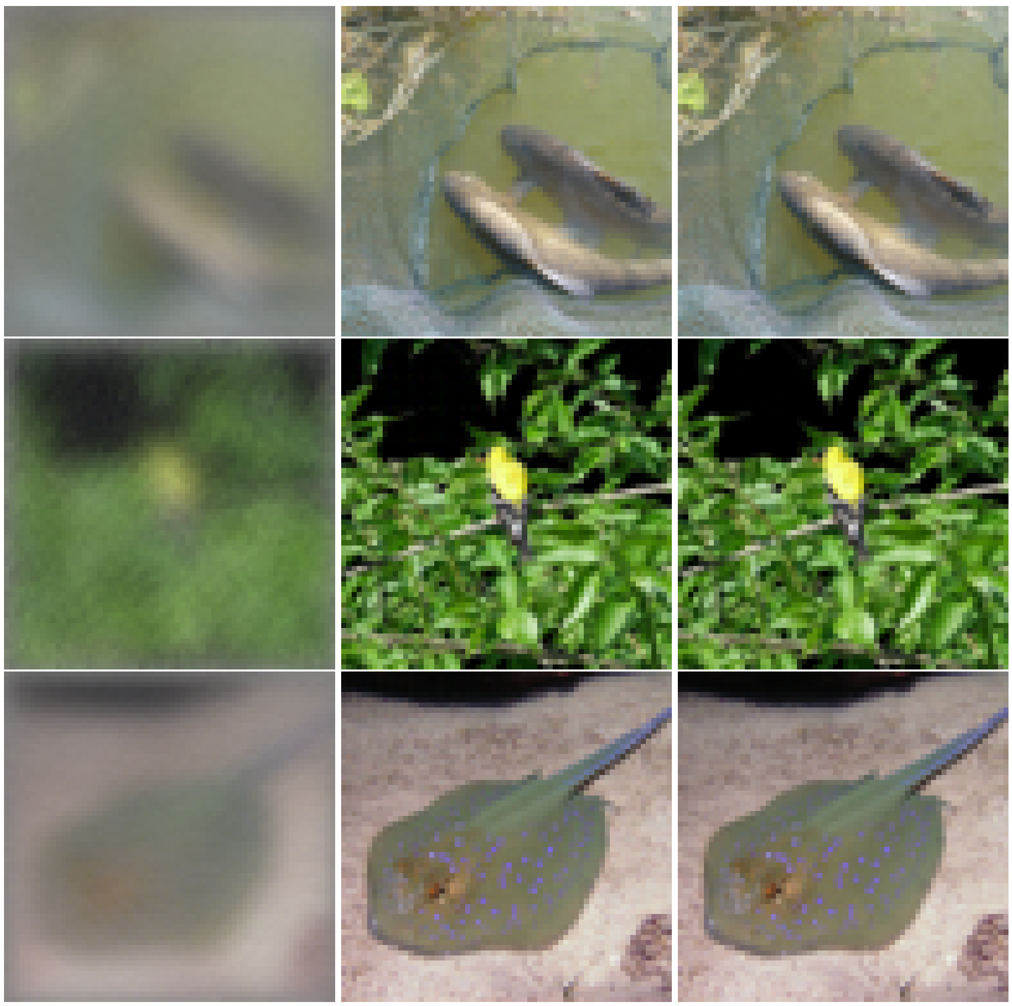

Figure 5. MAP-GA(D,NP) based image deblurring (with 7×7 uniform blur kernel) on ImageNet64. Left to right: observed, recovered, original images. We choose $num\_steps = 20$, $num\_iter = 50$.