Supplementary Material for Zero-Shot Adaptation for Approximate Posterior Sampling of Diffusion Models in Inverse Problems

Yaşar Utku Alçalar and Mehmet Akçakaya

University of Minnesota, Minneapolis
{alcal029,akcakaya}@umn.edu

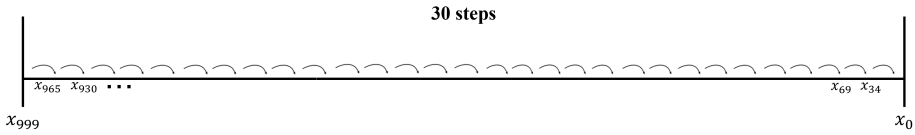
1 Implementation Details

1.1 Irregular Noise Schedules

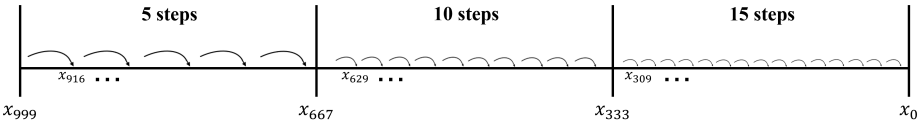
Sampling process for diffusion models can be accelerated via skipping some steps in the diffusion process [6, 7, 11]. A straightforward approach is to use uniformly spaced jumps across the noise schedule (see Fig. 1a) where the sampling path is uniformly spaced out by the selected number of steps in a regular manner. A schedule we commonly use in this study is a "$15, 10, 5$" schedule, which is pictorially depicted in Fig. 1b. This amounts to partitioning the total number of steps used in training into 3 parts and taking uniformly spaced $5, 10,$ and 15 samples from the respective segments, leading to increased sampling frequency at the lower noise levels. Although the samples are taken uniformly inside a given segment, each segment has different number of steps, making the whole schedule irregular (see Fig. 1b). We note that this irregular noise schedule is based on the ones proposed in [6], and the number of segments and the number of steps in each segment are chosen for the inverse problem setup based on computation time constraints while ensuring generalizability. We also note that a superior schedule may exist for a specific inverse problem, and optimization of these irregular noise schedules is an open problem to the best of our knowledge.

1.2 Model Details

Pre-trained models for FFHQ and ImageNet were taken from [1] and [6], respectively. Both score models were used without any retraining. For our approximation for the Hessian of the log prior, we utilized Daubechies 4 (db4) wavelet as the orthogonal wavelet transform. For our database evaluation, we employed 30 timesteps with "$15, 10, 5$" schedule for 10 epochs. Furthermore, for simplicity, we opted to initialize the learnable $\{\zeta_t\}$ and $\{\mathbf{D}_t\}$ values uniformly across steps and diagonals, respectively. For $\{\zeta_t\}$ initialization in Gaussian and motion blur, 0.2 was chosen. For random inpainting and super-resolution, 0.1 was used. For all inverse problem tasks, diagonals of $\{\mathbf{D}_t\}$ were initialized to 0.2. Adam optimizer with default settings was used.



(a) Fast sampling scheme that uses uniform jumps across the noise schedule.



(b) Fast sampling scheme that uses irregular jumps across the noise schedule.

Fig. 1: Illustrative figure for uniform/irregular noise schedules.

1.3 Baseline Implementations

DDRM. We followed the original implementation code provided by [9], and used the default values of $\eta = 0.85$ and $\eta_B = 1.0$ with 20 NFE DDIM.

Score-SDE, MCG, and DPS. For DPS implementation, we followed the original code provided by [2], while for MCG, we additionally performed projections onto the measurement set. For Score-SDE, we again employed projections onto the measurement set, without any gradient term to guide the diffusion process. We used 1000 NFE for each unless otherwise stated.

ΠGDM. We followed the original implementation detailed in [12] and used its public repository for implementation. We used 100 NFE unless otherwise stated.

All algorithms (including ZAPS) were implemented using a single NVIDIA A100 GPU. All algorithms used the same pre-trained unconditional diffusion models for a fair comparison.

1.4 Different Sampling Strategies

It is possible to use deterministic sampling schemes, such as denoising diffusion implicit models (DDIM) [11], to sample from a pre-trained DDPM model. Forward process for DDIM can be expressed as

$$q_\sigma(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \cdot q_\sigma(\mathbf{x}_t|\mathbf{x}_0)}{q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_0)}. \tag{1}$$

As evident from observation, each $\mathbf{x}_t$ is not solely dependent on $\mathbf{x}_{t-1}$ but also on $\mathbf{x}_0$, rendering the forward process non-Markovian. Given a noisy observation

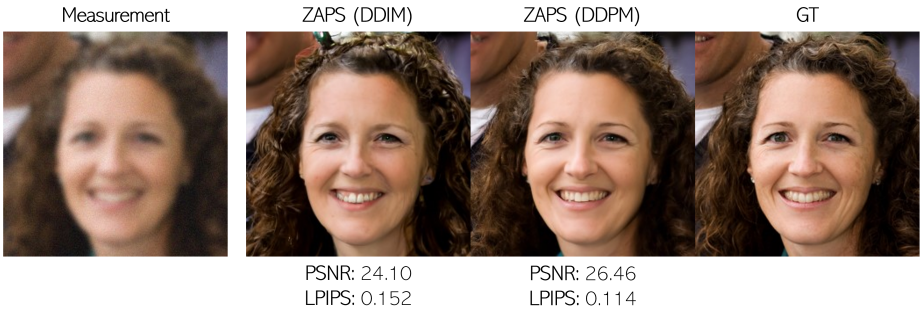


Fig. 2: Representative image reconstructed with ZAPS, using DDIM and DDPM sampling schemes. DDPM exhibits superior performance to DDIM in terms of both visual quality and quantitative metrics.

$\mathbf{x}_t$, the reverse process involves initially predicting the corresponding denoised $\mathbf{x}_0$ via Tweedie's formula

$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \cdot \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{2}$$

Using this estimate, one can generate a sample $\mathbf{x}_{t-1}$ from a sample $\mathbf{x}_t$ via:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \epsilon_\theta(\mathbf{x}_t, t) + \sigma_t \mathbf{z}, \tag{3}$$

where $\sigma_t = \eta \cdot \sqrt{\frac{(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)} \cdot \left(1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}\right)}$ and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. When $\sigma_t = 0$, sampling process becomes deterministic. We utilized both DDPM and DDIM with the same number of sampling steps within our ZAPS framework and as seen from Fig. 2, DDPM outperforms DDIM sampling both visually and quantitatively. Thus, we use DDPM in our work.

2 Additional Quantitative Results

2.1 ImageNet Results

Tab. 1 depicts quantitative evaluation of the state-of-the-art methods using LPIPS, SSIM, and PSNR for noisy inverse problems ($\sigma = 0.05$) on the ImageNet database. ZAPS shows competitive quantitative results either as the best or the second best among all the state-of-the-art methods.

2.2 Comparisons with DDNM and DiffPIR

We further compared ZAPS with the recently proposed DDNM [14] and DiffPIR [15] for Gaussian deblurring and super-resolution ($\times 4$) tasks (see Tab. 2). Each method is implemented using their respective public repository. ZAPS achieves $> 20\%$ improvement in terms of LPIPS, which is a perception-oriented metric.

Table 1: Quantitative results for Gaussian deblurring and super-resolution (×4) on ImageNet dataset. Best: **bold**, second-best: underlined. Comparison methods are omitted if they could not be implemented reliably for the given inverse problem task.

Method	Gaussian Deblurring			Super-Resolution (×4)		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
DPS [2]	0.230	0.668	22.16	0.275	0.673	21.77
MCG [3]	0.317	0.529	15.25	0.414	0.397	15.86
ΠGDM [12]	-	-	-	0.192	0.707	22.94
DDRM [9]	0.233	0.680	**23.34**	0.212	**0.725**	**24.33**
Score-SDE [1, 4, 13]	0.324	0.545	15.41	0.455	0.361	14.94
ZAPS **(Ours)**	**0.225**	**0.682**	22.45	**0.186**	0.718	23.82

2.3 Different Total Epochs - Timesteps Combinations with Fixed Total NFEs

As explained in our first ablation study in the main text, we also assessed the effectiveness of various combinations of total timesteps in the posterior sampling and number of epochs for fine-tuning quantitatively (see Tab. 3) while keeping the NFE constant. As anticipated, decreasing the number of epochs to 5 to allocate more timesteps had an adverse impact, where for some measurements, log-likelihood weights and approximated diagonals did not have the time to stabilize during the fine-tuning. Also as expected, 15 epochs × 20 timesteps combination and 10 epochs × 30 timesteps had similar outcomes, in which the latter outperformed the former slightly, and was used in the study.

3 Additional Qualitative Results

3.1 Effect of Using Distinct Weights $\{\zeta_t\}$

As part of our ablation studies, we examined the influence of selecting a shared weight ζ for every step versus using distinct weights ζ_t for each timestep. Fig. 3a shows that the shared ζ approach leads to artifacts that are highlighted in the

Table 2: Quantitative results for Gaussian deblurring and super-resolution (×4) on FFHQ dataset using NFE=100 ($\sigma = 0.05$) for each method. Best: **bold**, second-best: underlined.

Method	NFE↓	WCT(s)↓	Gaussian Deblur		SR (× 4)	
			LPIPS↓	PSNR↑	LPIPS↓	PSNR↑
DiffPIR [15]	100	12.47	0.182	25.86	0.143	26.02
DDNM [14]	100	11.88	0.172	**27.25**	0.148	26.76
ZAPS	100	10.85	**0.128**	26.41	**0.114**	**26.83**

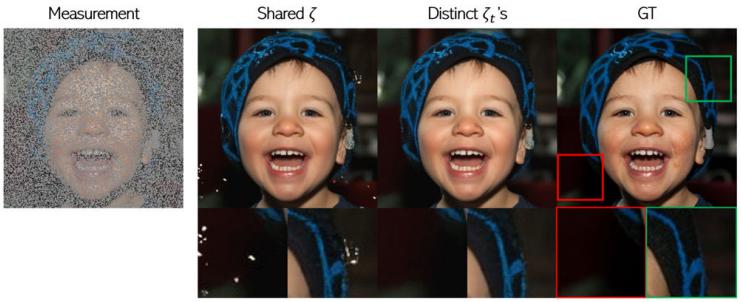
Table 3: Quantitative results for different epochs-steps combination (fixed NFE= 300) for the super-resolution ($\times 4, \sigma = 0.05$) inverse problem task on FFHQ dataset. Best: **bold**, second-best: underlined.

Combination	Schedule	LPIPS↓	SSIM↑	PSNR↑
15 epochs × 20 timesteps	"10, 7, 3"	**0.096**	0.763	26.48
10 epochs × 30 timesteps	"15, 10, 5"	0.104	**0.768**	**26.63**
5 epochs × 60 timesteps	"30, 20, 10"	0.109	0.741	26.39

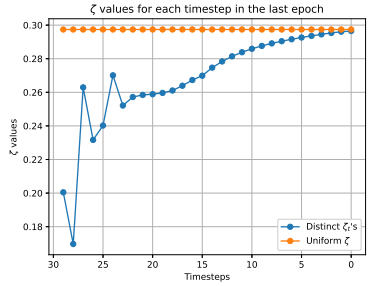
zoomed-in insets. Furthermore, Fig. 3b shows that the shared approach is susceptible to the goodness of the initialization, while the adaptive ζ_t weights are able to recover from arbitrary initializations. Fig. 3c further shows that the shared approach is prone to overfitting. Thus, the proposed approach with adaptive ζ_t log-likelihood weights is preferred.

3.2 Effect of Higher Number of Epochs for Fine-Tuning in ZAPS

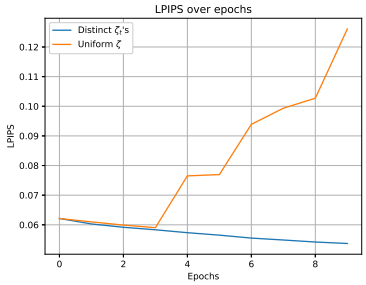
We evaluated our method on a representative ImageNet sample over 30 epochs for motion deblurring inverse problem task using 30 steps. As seen from Fig. 4, both the reconstruction faithfulness, and the visual quality, measured by PSNR



(a) Results for shared and distinct log-likelihood weights for irregular timesteps.



(b) Distinct ζ_t's recover from sub-optimal initialization.



(c) Shared ζ is prone to over-fitting.

Fig. 3: Study on different ζ choice strategies.

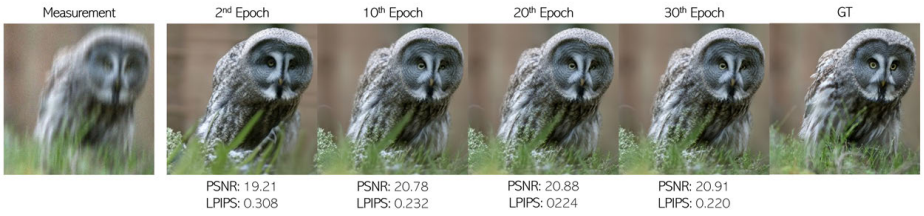


Fig. 4: Different epochs ZAPS reconstruction for motion deblurring task ($\sigma = 0.05$). Fine-tuning becomes redundant after the 10^{th} epoch when 30 sampling steps is being used.

and LPIPS respectively, demonstrate an increase via fine-tuning. However, after the 10^{th} epoch, the performance saturates and the gain through the log-likelihood weight update is diminished. Furthermore, no DIP-like overfitting was observed owing to the differences between the training loss function and the log-likelihood update term, as discussed in the main text. Thus, 10 epochs were used as the maximum number of epochs for the 30 step ZAPS setting to minimize total NFEs.

3.3 Effect of Wavelet Transform Choice

We further investigated using different types of orthogonal wavelets from the Daubechies wavelet family in Fig. 5. As seen from the results, the effect of the wavelet selection is negligible. Therefore, we opted to use Daubechies 4 wavelet as it is commonly used in sparse signal processing literature [10].

3.4 Additional Experimental Results

Further qualitative experimental results, comparing ZAPS with our state-of-the-art baseline, DPS, for various noisy inverse problem tasks ($\sigma = 0.05$) are

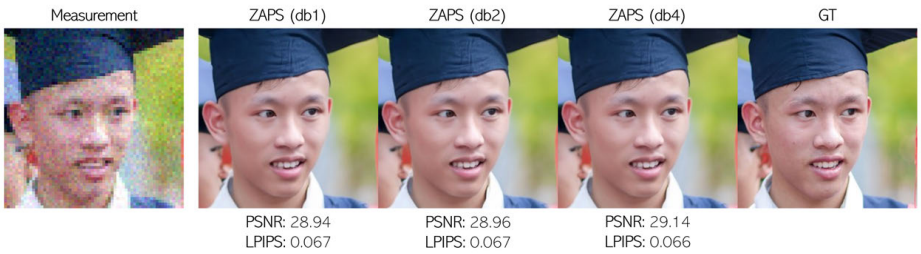


Fig. 5: Illustrative ZAPS results when different orthogonal wavelets are considered. The effect of the wavelet choice is trivial as each of them can converge to a good reconstruction. As it is commonly used in practice, we decide on using Daubechies 4 wavelet.

given in Figs. 6 to 11. We also provide inpainting task outcomes in Fig. 12 for various types of masks in addition to random and rectangular box inpainting. Our approach, involving the adjustment of the log-likelihood weights during fine-tuning and integration of irregular noise schedules with fewer sampling steps, results in a notable acceleration of approximately $\times 3$ on the FFHQ dataset and around $\times 4$ on ImageNet dataset, while also delivering superior performance.

4 Derivation of ΠGDM Update Using Woodbury Matrix Identity

For the log-likelihood estimation, ΠGDM [12] uses a Gaussian centered around $\mathbf{A}\hat{\mathbf{x}}_0$ to obtain the following score approximation:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \simeq \frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t} \mathbf{A}^\top (r_t^2 \mathbf{A}\mathbf{A}^\top + \sigma_y^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0). \tag{4}$$

In the text, we state that using Woodbury matrix identity, this can be rewritten as

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \simeq \frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t} (\mathbf{A}^\top \mathbf{A} + \eta \mathbf{I})^{-1} \mathbf{A}^\top (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0), \quad \text{where } \eta = \frac{\sigma_y^2}{r_t^2}, \tag{5}$$

which is more similar in form to DPS.

Proof. One can write Eq. (4) as

$$\frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t} \mathbf{A}^\top (r_t^2 \mathbf{A}\mathbf{A}^\top + \sigma_y^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0) = \frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t} \mathbf{A}^\top \frac{1}{r_t^2} (\mathbf{A}\mathbf{A}^\top + \eta \mathbf{I})^{-1} (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0), \tag{6}$$

where $\eta = \frac{\sigma_y^2}{r_t^2}$. Applying Woodbury matrix identity, $(\mathbf{A}\mathbf{A}^\top + \eta \mathbf{I})^{-1}$ can be rewritten as

$$(\mathbf{A}\mathbf{A}^\top + \eta \mathbf{I})^{-1} = \frac{\mathbf{I}}{\eta} - \frac{\mathbf{A}}{\eta} \left(\mathbf{I} + \frac{1}{\eta} \mathbf{A}^\top \mathbf{A} \right)^{-1} \frac{\mathbf{A}^\top}{\eta} \tag{7}$$

$$= \frac{1}{\eta} \left(\mathbf{I} - \mathbf{A} \left(\eta \mathbf{I} + \mathbf{A}^\top \mathbf{A} \right)^{-1} \mathbf{A}^\top \right). \tag{8}$$

Thus, Eq. (6) becomes

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \propto \frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t} \frac{\mathbf{A}^\top}{\eta} \left(\mathbf{I} - \mathbf{A} \left(\eta \mathbf{I} + \mathbf{A}^\top \mathbf{A} \right)^{-1} \mathbf{A}^\top \right) (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0). \tag{9}$$

Noting $\mathbf{I} = (\mathbf{A}^\top \mathbf{A} + \eta \mathbf{I})(\mathbf{A}^\top \mathbf{A} + \eta \mathbf{I})^{-1}$ yields

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \propto \frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t} \mathbf{I} (\mathbf{A}^\top \mathbf{A} + \eta \mathbf{I})^{-1} \mathbf{A}^\top (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0), \tag{10}$$

which is similar to the DPS update, where $(\mathbf{A}^\top \mathbf{A} + \eta \mathbf{I})^{-1}$ is the difference.

References

1. Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S.: Ilvr: Conditioning method for denoising diffusion probabilistic models. in 2021 ieee. In: CVF international conference on computer vision (ICCV). pp. 14347–14356 (2021)
2. Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion posterior sampling for general noisy inverse problems. International Conference on Learning Representations (2023)
3. Chung, H., Sim, B., Ryu, D., Ye, J.C.: Improving diffusion models for inverse problems using manifold constraints. Advances in Neural Information Processing Systems (2022)
4. Chung, H., Sim, B., Ye, J.C.: Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
7. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022)
8. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4396–4405 (2019)
9. Kawar, B., Elad, M., Ermon, S., Song, J.: Denoising diffusion restoration models. In: Advances in Neural Information Processing Systems (2022)
10. Lustig, M., Donoho, D., Pauly, J.M.: Sparse mri: The application of compressed sensing for rapid mr imaging. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine **58**(6), 1182–1195 (2007)
11. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. International Conference on Learning Representations (2020)
12. Song, J., Vahdat, A., Mardani, M., Kautz, J.: Pseudoinverse-guided diffusion models for inverse problems. In: International Conference on Learning Representations (2022)
13. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. International Conference on Learning Representations (2020)
14. Wang, Y., Yu, J., Zhang, J.: Zero-shot image restoration using denoising diffusion null-space model. The Eleventh International Conference on Learning Representations (2023)
15. Zhu, Y., Zhang, K., Liang, J., Cao, J., Wen, B., Timofte, R., Gool, L.V.: Denoising diffusion models for plug-and-play image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (NTIRE) (2023)

Fig. 6: Gaussian deblurring results for ZAPS and DPS on FFHQ [8] 256×256 dataset.

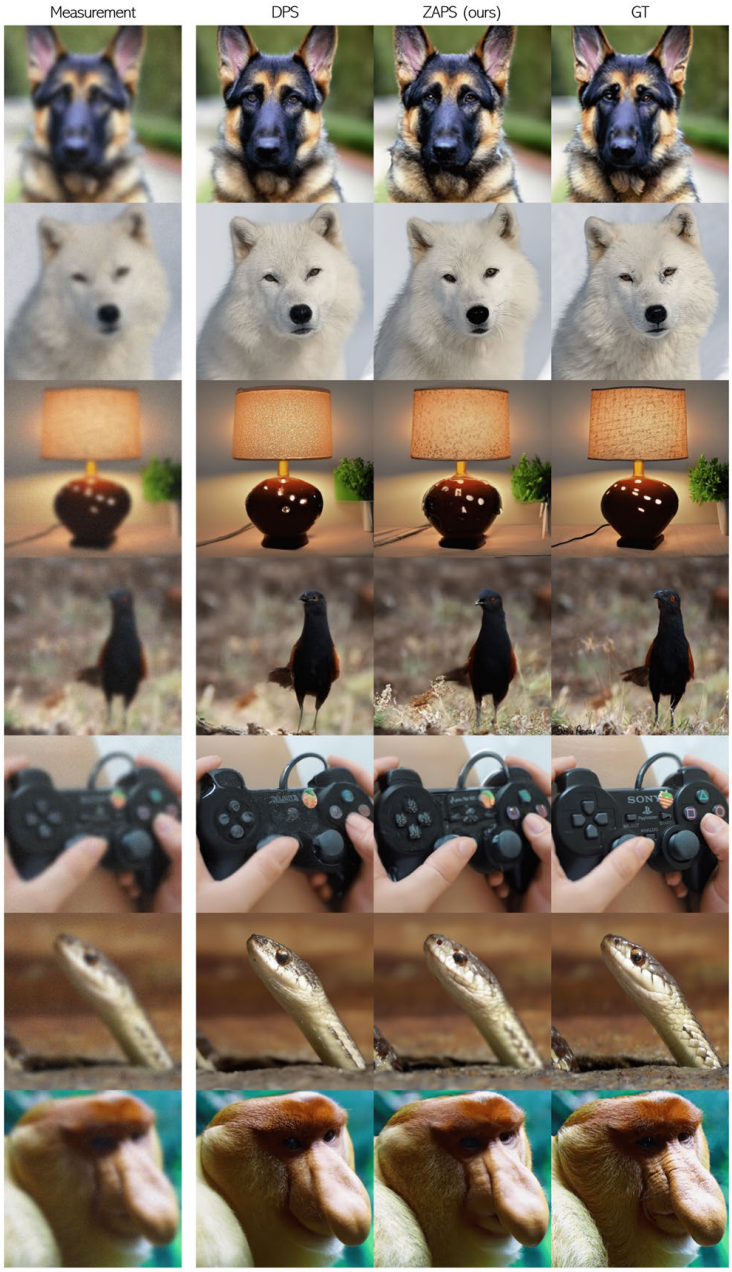


Fig. 7: Gaussian deblurring results for ZAPS and DPS on ImageNet [5] 256×256 dataset.

Fig. 8: Super-resolution (×4) results for ZAPS and DPS on FFHQ [8] 256×256 dataset.

Fig. 9: Super-resolution (×4) results for ZAPS and DPS on ImageNet [5] 256×256 dataset.

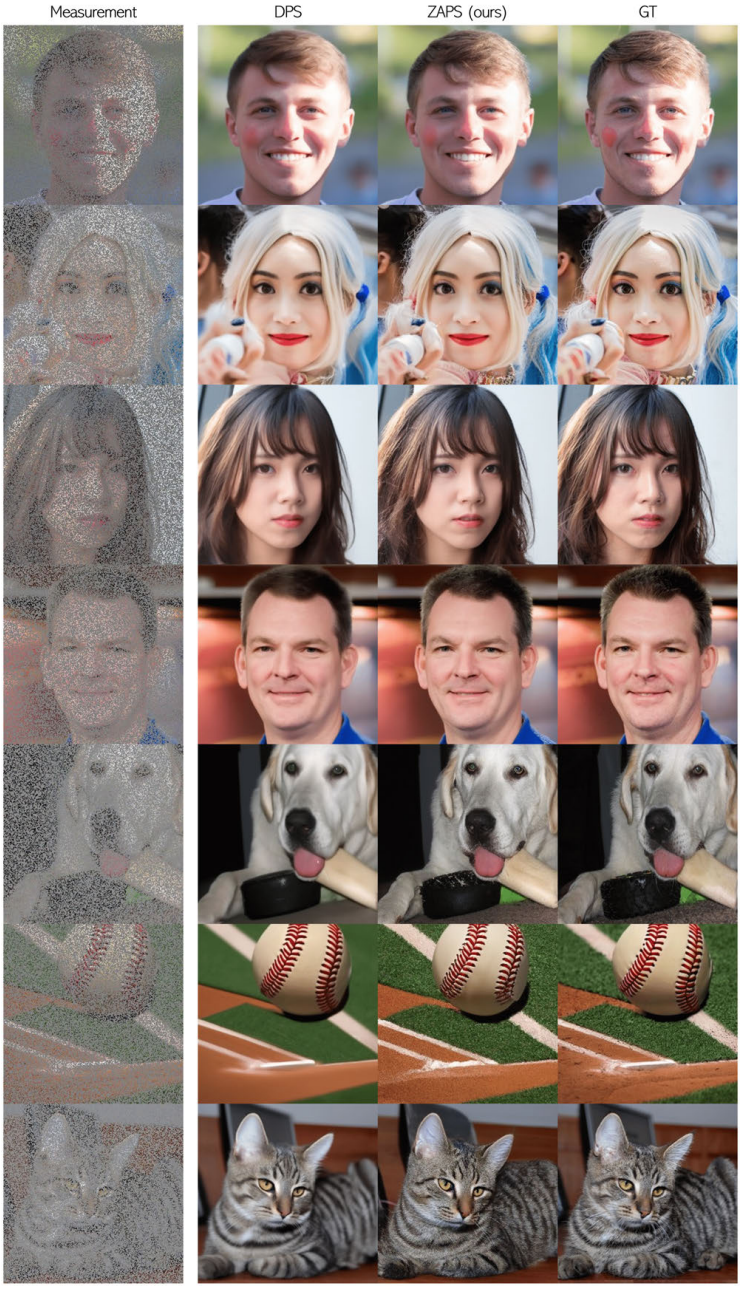


Fig. 10: Random inpainting (70%) results for ZAPS and DPS on FFHQ [8] 256×256 and ImageNet [5] 256×256 dataset.

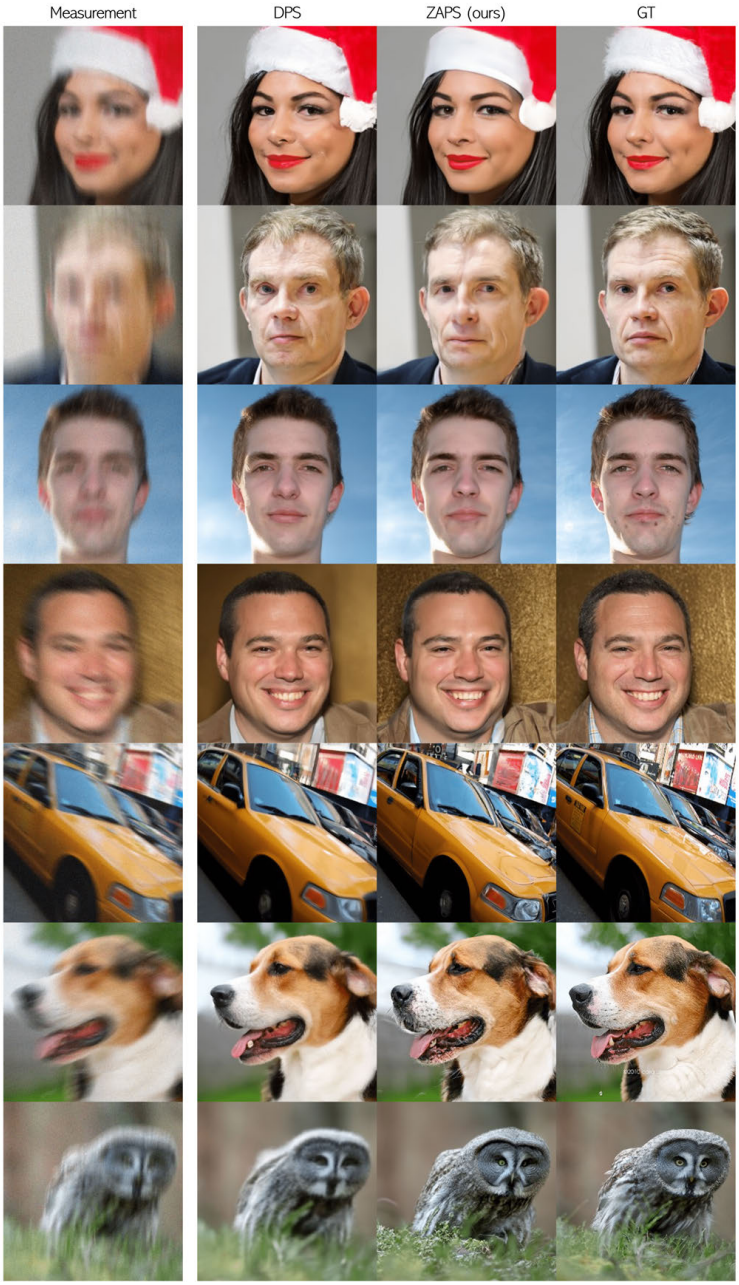


Fig. 11: Motion deblurring results for ZAPS and DPS on FFHQ [8] 256×256 and ImageNet [5] 256×256 dataset.

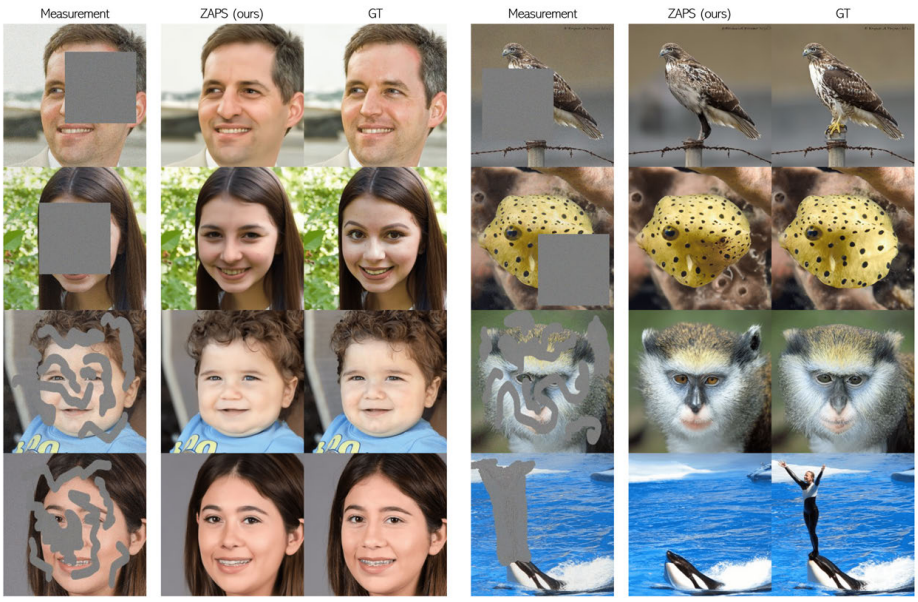


Fig. 12: Representative ZAPS reconstructions for image inpainting task using different masks (box size is 128×128) with $\sigma = 0.05$.