

Iterative Renoising as an Efficient Way to Solve Imaging Inverse Problems with Diffusion

Matt Bendel, Saurav Shastri, Rizwan Ahmad, and Phil Schniter
Supported by NIH grant R01-EB029957



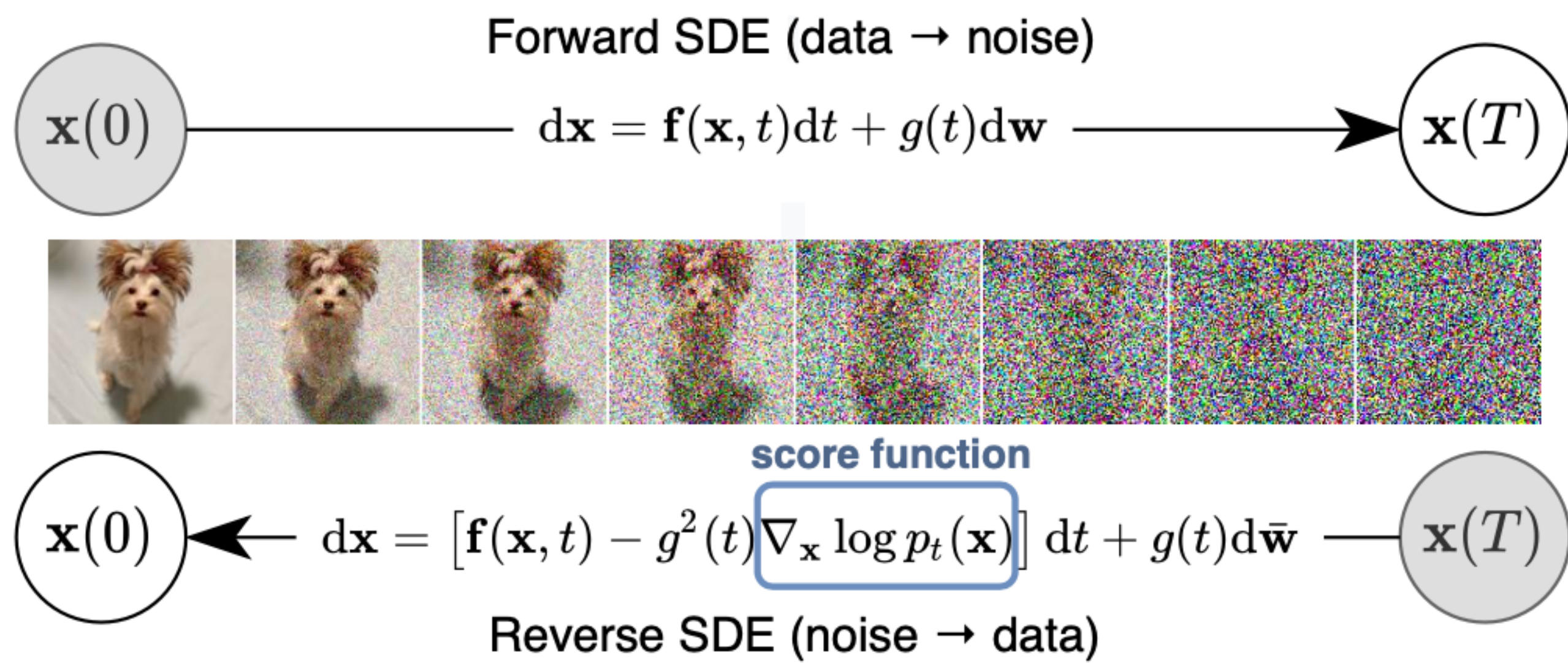
THE OHIO STATE UNIVERSITY

2025 BASP Frontiers Conference
Villars-sur-Ollon, Switzerland

Imaging inverse problems

- Measurements $\mathbf{y} = \mathcal{A}(\mathbf{x}_0)$ of unknown image $\mathbf{x}_0 \sim p(\mathbf{x})$
 - $\mathcal{A}(\cdot)$ masks, distorts, and/or corrupts $\mathbf{x}_0$ with noise.
 - Examples: denoising, deblurring, inpainting, super-resolution, phase retrieval, computed tomography (CT), magnetic resonance imaging (MRI), etc.
- Goal: accurately sample from the posterior $p(\mathbf{x}|\mathbf{y})$
- Challenges
 - Ill-posed: Many hypotheses of $\mathbf{x}_0$ can explain $\mathbf{y}$
 - High dimensional

Diffusion methods



- Diffusion methods are powerful ways to sample from a complex distribution $p(\mathbf{x})$
- The forward process **gradually adds noise** to $\mathbf{x}(0) \sim p(\mathbf{x})$. The reverse process **gradually denoises** pure noise $\mathbf{x}(T)$, eventually generating a sample from $p(\mathbf{x})$
- To discretize, we'll assume VP DDPM, where step $t \in \{1, \dots, T\}$ provides

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

- Although the reverse process is usually written using the **score function** $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$, it can also be written using the **MMSE denoiser** $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t\}$ via Tweedie's rule

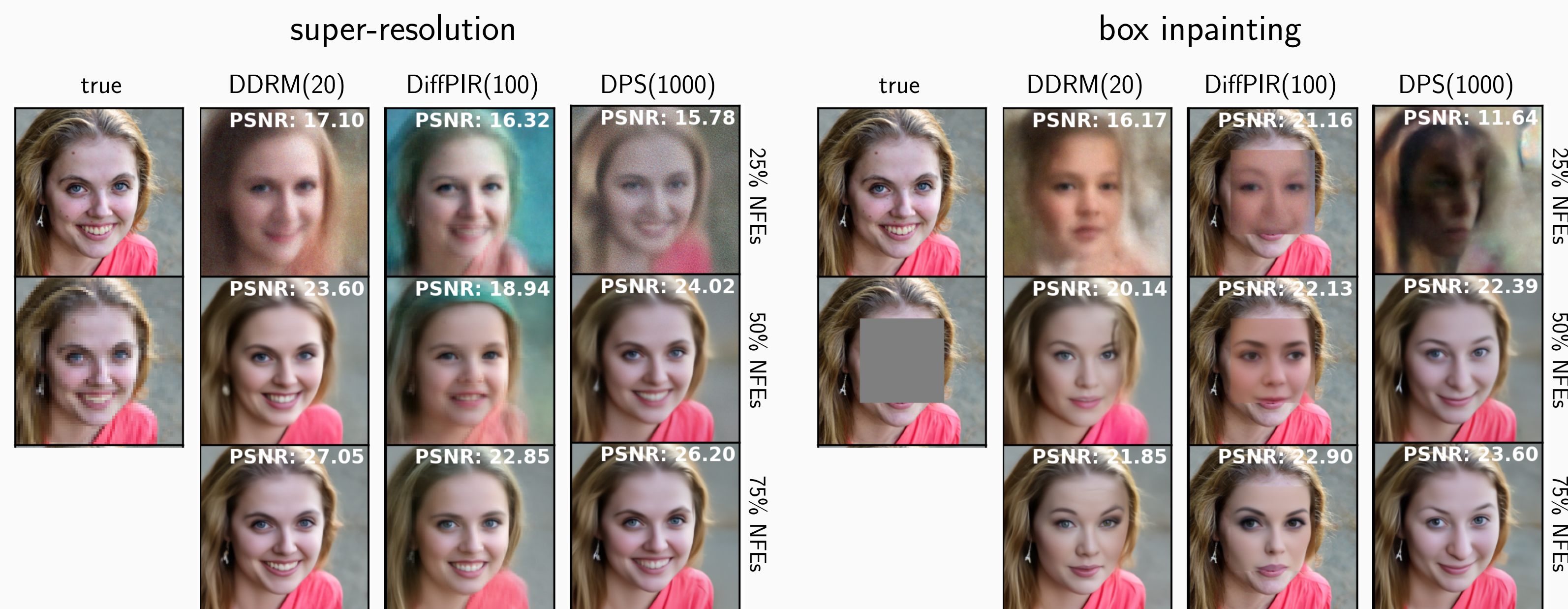
$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) = \frac{\sqrt{\alpha_t} \mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t\} - \mathbf{x}_t}{1 - \alpha_t}$$

Solving inverse problems with diffusion

- We can train an approximation to the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ from samples of $\mathbf{x}_0 \sim p(\mathbf{x})$, independently of any forward model $\mathcal{A}(\cdot)$
- Then we can run the diffusion reverse process using an approximation to the *conditional* score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{y})$ in place of $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$, or equivalently by using $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$ in place of $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t\}$
- Many strategies have been proposed to approximate $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{y})$ or $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$ using some combination of a pretrained approximation to $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ and the likelihood function $p(\mathbf{y}|\mathbf{x}_0)$
 - Popular methods include DDRM [1], DPS [2], DDNM [3], IIGDM [4], DiffPIR [5], DDS [6], etc.

The main challenge

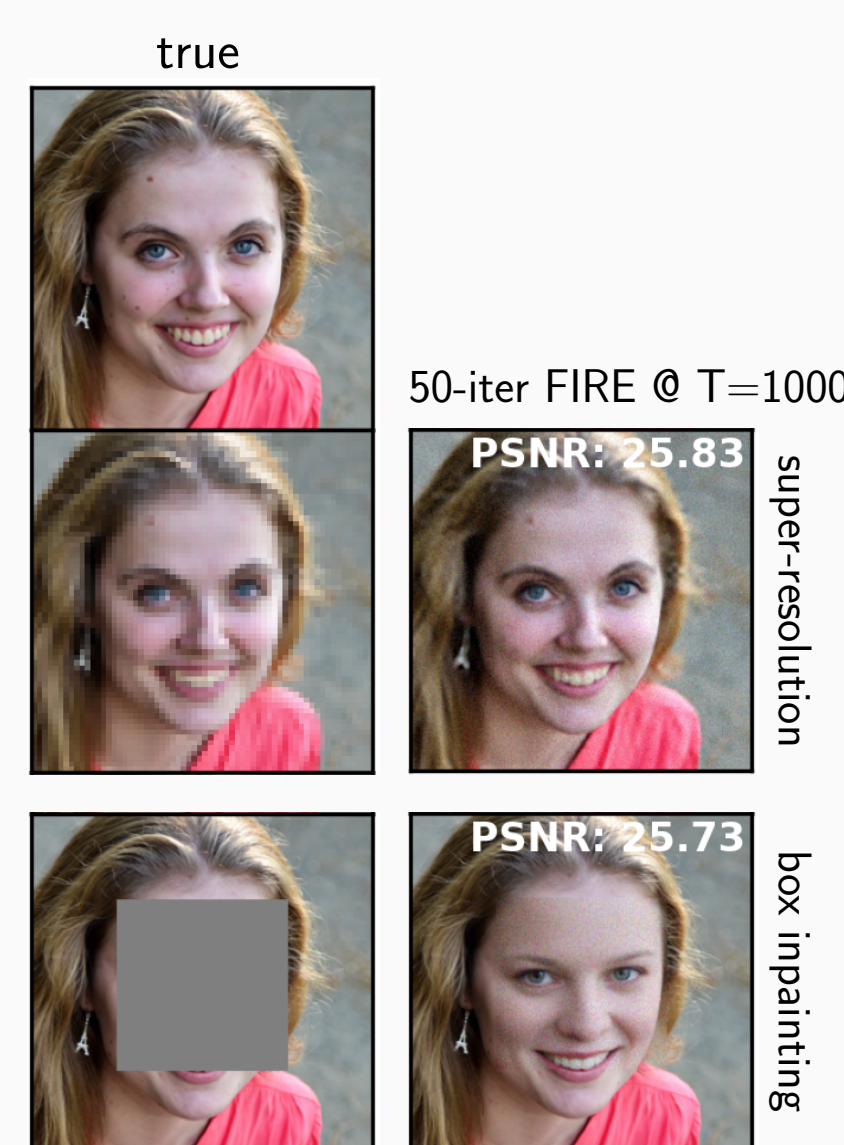
- The main challenge is computing a good approximation to $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$ under computational constraints
- Any given method can be assessed by computing the MSE of its $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$ approximation, since the exact $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$ minimizes MSE



- The DDRM, DiffPIR, and DPS approximations of $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$ are all relatively poor, especially early in the reverse process

Fast Iterative REnoising (FIRE)

- We propose an iterative approximation of $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$ that we call **FIRE**
- For linear inverse problems $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{w}$ with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I})$, FIRE iterates the following steps after initializing $\mathbf{r} = \mathbf{x}_t$ and $\sigma_r^2 = (1 - \alpha_t)/\alpha_t$:
 - $\tilde{\mathbf{x}}_0 \leftarrow \text{Denoise}(\mathbf{r}; \sigma_r^2), \quad \sigma_r^2 \leftarrow \mathbb{E}\|\tilde{\mathbf{x}}_0 - \mathbf{x}_0\|^2/d$
 - $\hat{\mathbf{x}}_0 \leftarrow \arg \min_{\mathbf{x}} \frac{1}{\sigma_w^2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \frac{1}{\sigma_r^2} \|\tilde{\mathbf{x}}_0 - \mathbf{x}\|^2$
 - $\sigma_r^2 \leftarrow \sigma_r^2/\rho$ for some $\rho > 1$... decrease denoising variance
 - renoise**: set $[\mathbf{r} \leftarrow \hat{\mathbf{x}}_0 + \mathbf{n}]$ with **colored Gaussian $\mathbf{n}$ that gives $\mathbf{r} \sim \mathcal{N}(\mathbf{x}_0, \sigma_r^2 \mathbf{I})$**
- Key idea: **renoising ensures that the denoiser sees AWGN**, consistent with how it was trained!
- Two options for renoising: exact SVD-based, or approximate $(\mathbf{A}, \mathbf{A}^\top)$ -based



DDfire: Putting the FIRE into diffusion

- FIRE can be plugged into any diffusion reverse process. We use DDPM
- Because FIRE uses multiple NFEs per $\mathbb{E}\{\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}\}$, we subsample the diffusion steps $\{t\}$ using DDIM and **schedule the FIRE iterations** to meet a given total-NFE budget
- We allocate FIRE iterations using a **“waterfilling”** procedure, which is best illustrated using *inverse* variances, i.e., precisions:
- Basically, waterfilling ensures that FIRE's final-iteration denoiser-input-precision meets a target at each DDIM step
- The resulting **“DDfire”** approach outperforms many state-of-the-art diffusion methods at the tested NFE budgets of 20, 100, 1000

Noisy FFHQ results

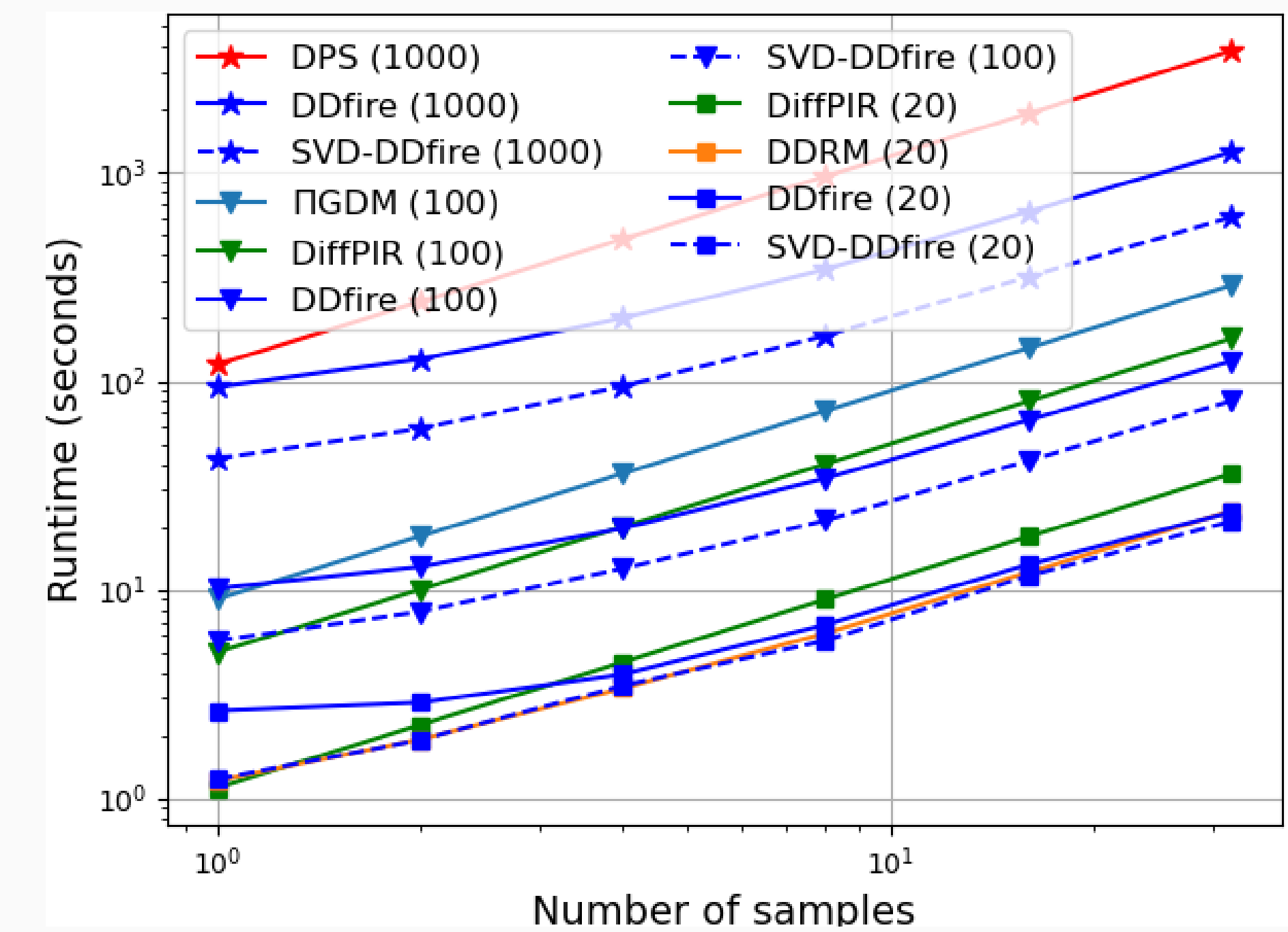
- Results on 256×256 FFHQ faces with measurement noise $\sigma_w = 0.05$:

#	NFEs	Model	Inpaint (box)			Deblur (Gaussian)			Deblur (Motion)			4× Super-resolution		
			PSNR↑	LPIPS↓	FID↓	PSNR↑	LPIPS↓	FID↓	PSNR↑	LPIPS↓	FID↓	PSNR↑	LPIPS↓	FID↓
20	DiffPIR		20.87	0.2741	41.50	23.55	0.3269	41.29	27.31	0.2704	29.27	22.32	0.3560	44.85
	DDRM		22.02	0.2052	40.61	26.27	0.2896	51.70	-	-	-	28.62	0.2417	45.82
	DDfire		21.80	0.1974	28.49	27.18	0.2843	36.22	28.52	0.2455	28.86	27.02	0.2917	37.72
100	DiffPIR		22.44	0.2415	31.98	24.57	0.2936	34.82	26.91	0.2683	26.67	26.76	0.3061	32.33
	ITGDM		21.75	0.2614	44.41	24.34	0.3125	45.34	25.94	0.2706	41.95	25.42	0.3109	51.41
	DDfire		23.78	0.1623	26.75	27.48	0.2274	25.48	27.79	0.2193	25.91	27.20	0.2399	26.24
1000	DPS		22.84	0.1793	35.69	26.32	0.2327	25.18	27.64	0.2176	27.17	27.11	0.2360	27.38
	DDfire		24.14	0.1579	24.56	26.84	0.2259	24.68	27.71	0.2155	24.57	27.32	0.2356	25.75

- DDfire **outperforms the competitors in 33 of the 36 cases**

Fast inference speed

- DDfire can be run using an **SVD or CG**, and in **batch mode**



- SVD-DDfire **faster than all competitors** when batch size ≥ 2 (10x faster than DPS)

Extension to generalized-linear models

- To handle problems like phase retrieval, dequantization, Poisson regression, and non-Gaussian additive noise, we leverage to the **generalized linear model (GLM)**:

$$\mathbf{y} \sim p(\mathbf{y}|\mathbf{z}_0) = \prod_{j=1}^m p_{y|z}(y_j|z_{0,j}) \quad \text{with} \quad \mathbf{z}_0 \triangleq \mathbf{A}\mathbf{x}_0$$

- GLM-FIRE is constructed with **expectation-propagation** iterations between linear FIRE and a scalar MMSE inference stage:

Noisy FFHQ phase-retrieval results

- Results on 256×256 FFHQ faces with shot noise level α_{shot} :

# NFEs	Model	4x OSF @ $\alpha_{\text{shot}} = 8$			4x CDP @ $\alpha_{\text{shot}} = 45$		
		PSNR $\uparrow$	LPIPS $\downarrow$	FID $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$	FID $\downarrow$
-	HIO	23.66	0.5299	130.58	17.59	0.5818	84.87
1000	DOLPH [7]	14.73	0.7089	389.88	25.76	0.2163	32.93
1000	DPS [2]	23.63	0.3326	53.91	29.19	0.1994	27.87
800	prDeep [8]	30.90	0.1585	31.51	19.24	0.4352	59.44
800	GLM-DDfire	33.56	0.1160	28.94	30.01	0.1767	23.49
100	GLM-DDfire	25.88	0.2643	46.54	30.16	0.1707	23.30

- GLM-DDfire gives **state-of-the-art noisy phase-retrieval** performance with both oversampled Fourier (OSF) and coded diffraction pattern (CDP) operators

References

- B. Kavar, M. Elad, S. Ermon, and J. Song, “Denoising diffusion restoration models,” in *Proc. Neural Info. Process. Syst. Conf.*, 2022.
- H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, “Diffusion posterior sampling for general noisy inverse problems,” in *Proc. Intl. Conf. Learn. Rep.*, 2023.
- Y. Wang, J. Yu, and J. Zhang, “Zero-shot image restoration using denoising diffusion null-space model,” in *Proc. Intl. Conf. Learn. Rep.*, 2023.
- J. Song, A. Vahdat, M. Mardani, and J. Kautz, “Pseudoinverse-guided diffusion models for inverse problems,” in *Proc. Intl. Conf. Learn. Rep.*, 2023.
- Y. Zhu, K. Zhang, J. Liang, J. Cao, B. Wen, R. Timofte, and L. Van Gool, “Denoising diffusion models for plug-and-play image restoration,” in *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, pp. 1219–1229, 2023.
- H. Chung, S. Lee, and J. C. Ye, “Decomposed diffusion sampler for accelerating large-scale inverse problems,” in *Proc. Intl. Conf. Learn. Rep.*, 2024.
- S. Shoushtari, J. Liu, and U. S. Kamilov, “Diffusion models for phase retrieval in computational imaging,” in *Proc. Asilomar Conf. Signals Syst. Comput.*, pp. 779–783, 2023.
- C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, “prDeep: Robust phase retrieval with flexible deep neural networks,” in *Proc. Intl. Conf. Mach. Learn.*, pp. 3501–3510, 2018.