Computational Imaging and Diffusion Project

Stephone Christian

Stanford University

Motivation

Diffusion models provide strong data-driven priors by learning the score $\nabla_{x_t} \log p_t(x_t)$ over the data distribution. As shown in [3], they map a forward diffusion process that gradually adds noise to data onto a reverse process that iteratively denoises. The forward process is

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z, \text{ where } z \sim \mathcal{N}(0,I); \quad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i; \quad \alpha_t = 1 - \beta_t$$

This lets pretrained diffusion models be used as priors for inverse problems without task-specific training. Methods like SDEdit [1] and ScoreALD [2] integrate measurement information into the reverse diffusion to produce samples consistent with both the learned prior and the observations.

We study how such methods perform on **inpainting** and **deconvolution**, using a **pretrained DDPM on Flickr-Faces-HQ (FFHQ)** as a prior. The goal is to compare how different conditioning strategies—simple partial noising (SDEdit), annealed Langevin dynamics (ScoreALD), and normalized likelihood gradients (DPS)—affect fidelity to measurements, perceptual quality, and robustness.

Challenges and Limitations

- Computational cost. All methods use iterative reverse diffusion (~1000 steps), which is slow compared to one-shot neural networks. Each step needs a full UNet forward pass, and GPU compute (e.g., T4) is typically required for practical runtime.
- Domain mismatch. The FFHQ-pretrained model is optimized for faces. Reconstructions are biased toward face-like structure, so performance degrades on other image types.
- Metric gaps. PSNR and LPIPS can conflict: higher PSNR does not always correspond to better perceptual quality, and LPIPS can favor sharper but less data-consistent images

References

[1]Meng, C., He, Y., et. al. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. *International Conference on Learning Representations (ICLR) (2022).*

[2] Song, Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations. *International Conference on Learning Representations (ICLR) (2021)*

[3] Ho, J., Jain, et al. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems (NeurIPS) (2020)*

SDEdit

Algorithm 1 Guided image synthesis and editing with SDEdit (VE-SDE)

Require: $\mathbf{x}^{(g)}$ (guide), t_0 (SDE hyper-parameter), N (total denoising steps)
$\Delta t \leftarrow \frac{t_0}{N}$
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\mathbf{x} \leftarrow \mathbf{x} + \sigma(t_0)\mathbf{z}$
for $n \leftarrow N$ **to** 1 **do**
  $t \leftarrow t_0 \frac{n}{N}$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\epsilon \leftarrow \sqrt{\sigma^2(t) - \sigma^2(t - \Delta t)}$
  $\mathbf{x} \leftarrow \mathbf{x} + \epsilon^2 \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}, t) + \epsilon \mathbf{z}$
end for
Return $\mathbf{x}$

ScoreALD

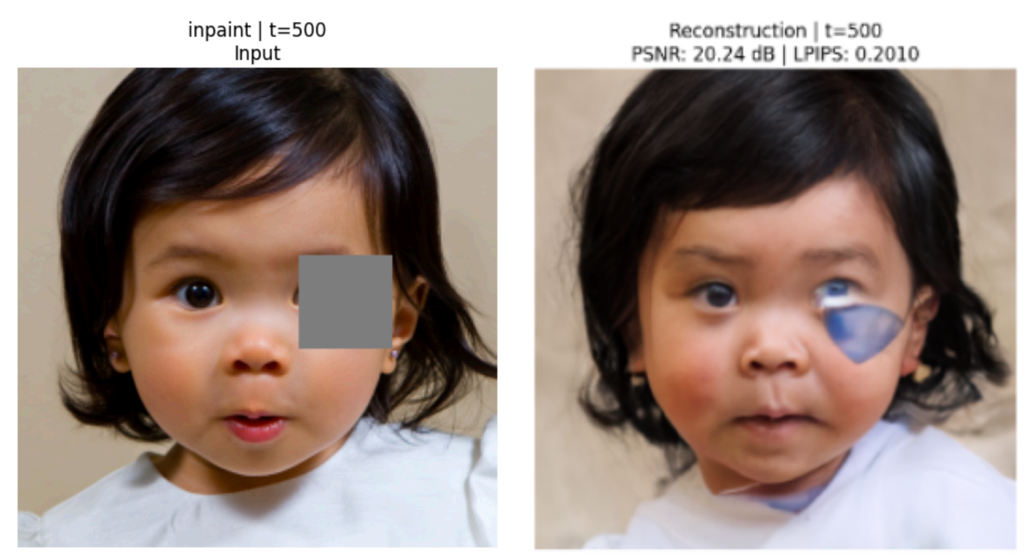
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t + (1-\bar{\alpha}_t)\boldsymbol{s}_\theta(x_t, t))$
  $\boldsymbol{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_0 + \sqrt{1-\alpha_t}\boldsymbol{z}$
  $\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1} - \frac{1}{2(\sigma^2 + \gamma_t^2)} \nabla_{\boldsymbol{x}_t} \left\| \mathcal{A}(\boldsymbol{x}_t) - \boldsymbol{y} \right\|^2$
end for
return $\mathbf{x}_0$

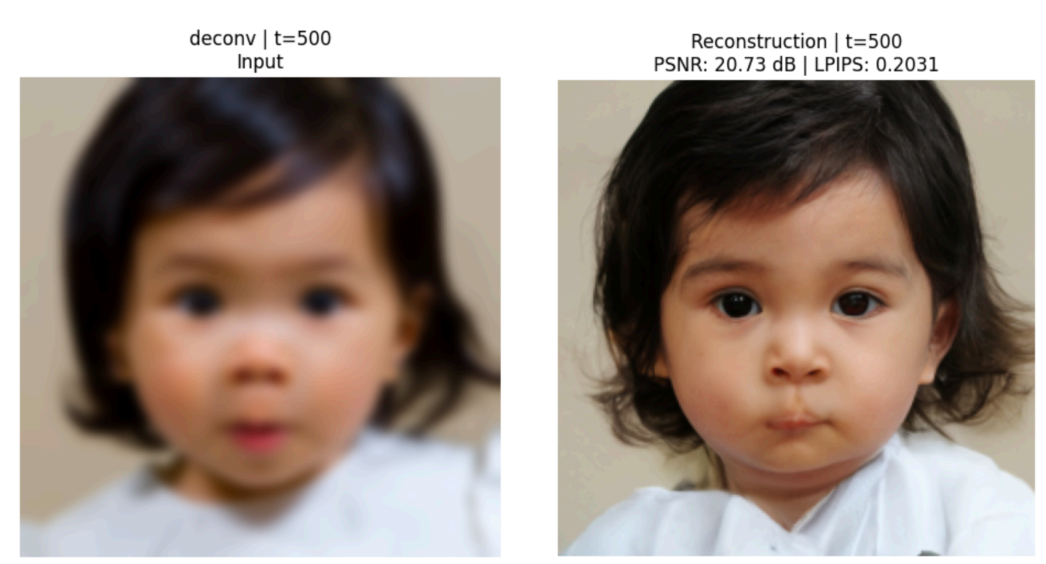
DPS

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t + (1-\bar{\alpha}_t)\boldsymbol{s}_\theta(x_t, t))$
  $\boldsymbol{x}'_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_0 + \sqrt{1-\alpha_t}\boldsymbol{z}$
  $\boldsymbol{x}_{t-1} = \boldsymbol{x}'_{t-1} - \zeta_t \nabla_{\boldsymbol{x}_t} \left\| \mathcal{A}(\hat{\boldsymbol{x}}_0) - \boldsymbol{y} \right\|^2$
end for
return $\mathbf{x}_0$

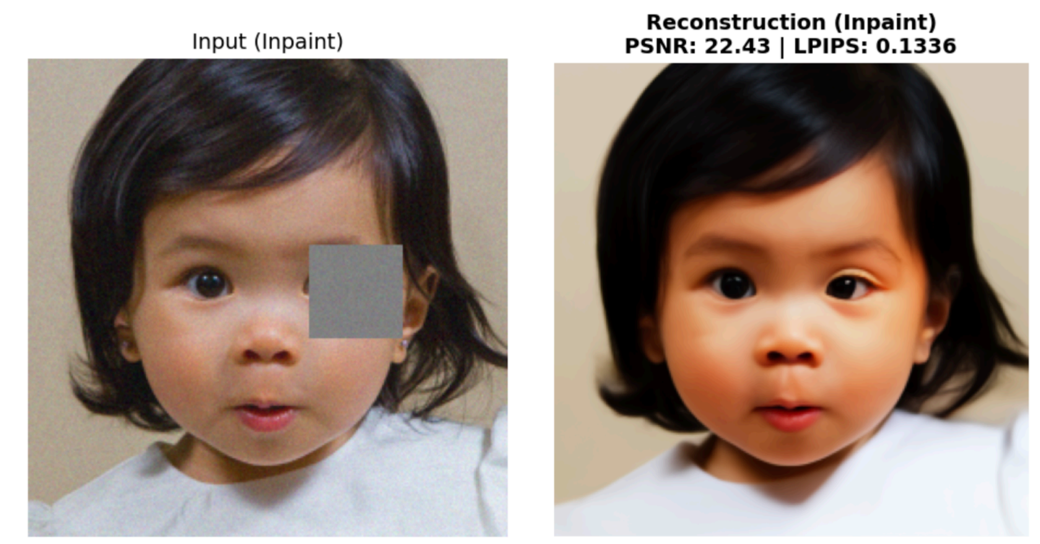
In prac

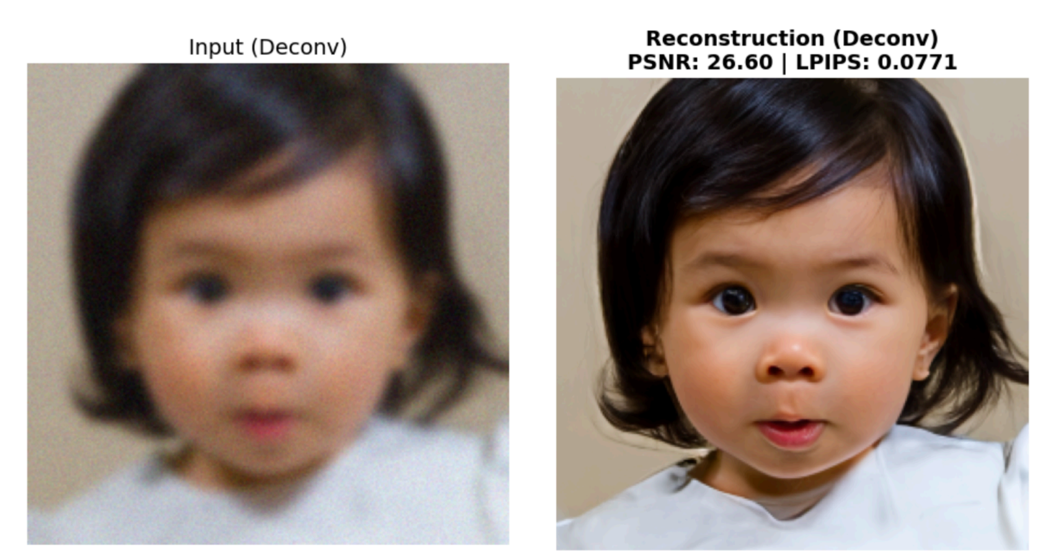
Score Distillation Editing (SDEdit)





Score Based Annealed Langevine Dynamics (ScoreALD)





Diffusion Posterior Sampling (DPS)