Analysis of Diffusion Models in Inverse Problems

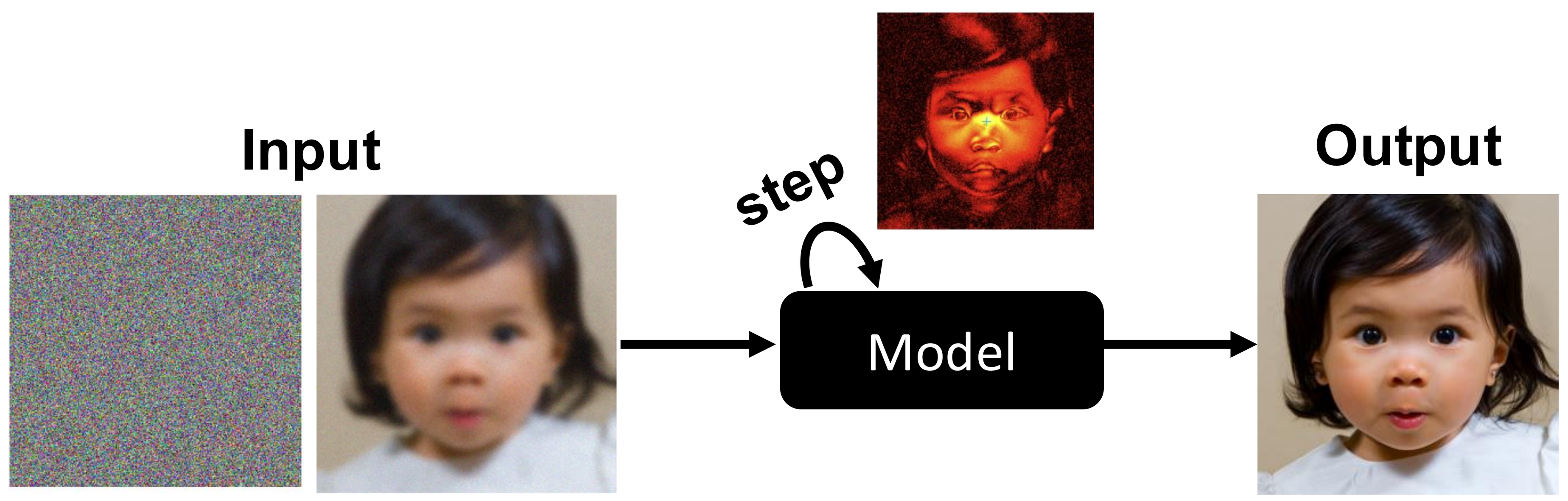
Munir Bshara

Stanford University

EE367 Final Project, Winter 2026

Motivation

- Imperfect yet **structured** observations
- Diffusion models can learn these complex structures
- Inverse problems require ways to specify “what is?” a good solution
- Can diffusion models **implicitly regularize?**



- **What is going on behind the hood?**

Related Work

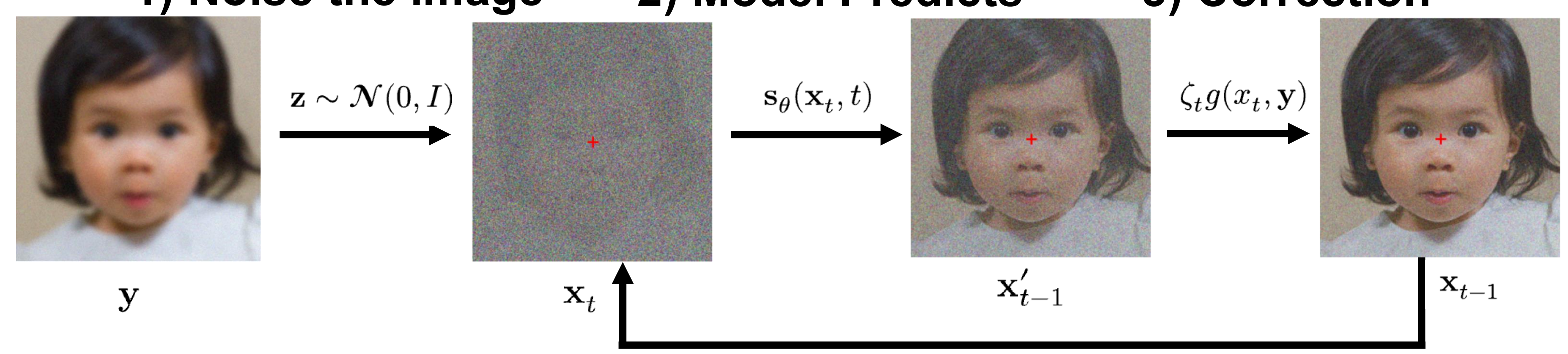
- Classical inverse problem techniques based in optimization: ADMM [4]
- Neural networks have been shown to be compressible [6]
- Diffusion models are data driven [1]

References

[1] Song et. al, “Score-Based Generative Modeling through Stochastic Differential Equations,” *ICLR,* 2021.
[2] Chung et. al, “Diffusion Posterior Sampling for General Noisy Inverse Problems,” *ICLR*, 2023
[3] Ventura et. al, “Manifolds, Random Matrices and Spectral Gaps: The Geometric Phases of Generative Diffusion,” *arXiv:2410.05898*, 2024
[4] Boyd et. al, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” FTML, 2011
[5] M. A. Turk et al, “Face recognition using eigenfaces”, IEEE, 1991
[6] Han et. al, “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,” ICLR, 2016

Methods

1) Noise the image **2) Model Predicts** **3) Correction**



Update Equations

$$\mathbf{x}'_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t + (1-\alpha_t)\mathbf{s}_\theta(\mathbf{x}_t, t))$$

$$\mathbf{x}_{t-1} = \mathbf{x}'_{t-1} + \zeta_t g(x_t, \mathbf{y})$$

SDEdit

- No correction

$$\zeta_t g(\mathbf{x}_t, \mathbf{y}) = 0$$

ScoreALD

- Likelihood correction

$$\zeta_t = \frac{1}{\sigma^2 + \gamma_t^2}, g(\mathbf{x}_t, \mathbf{y}) = -\nabla_{\mathbf{x}_t}\|\mathbf{y} - A(\mathbf{x})\|_2^2$$

DPS

- Stabilized likelihood correction

$$\zeta_t = \frac{\zeta}{\|\mathbf{y} - A(\mathbf{x})\|_2}, g(\mathbf{x}_t, \mathbf{y}) = -\nabla_{\mathbf{x}_t}\|\mathbf{y} - A(\mathbf{x})\|_2^2$$

Methods of Analysis

Score Jacobian: $J_s e_i = \frac{\partial \mathbf{s}_\theta(\mathbf{x}_t, t)}{\partial [\mathbf{x}_t]_i}$

- Measures affect of perturbing pixel i

Reverse Step Response: $J_f e_i = \frac{1}{\sqrt{\alpha_t}}(e_i + (1-\alpha_t)J_s e_i)$

- Spatial footprint of perturbation after one inner iteration

Top k Singular Vectors:

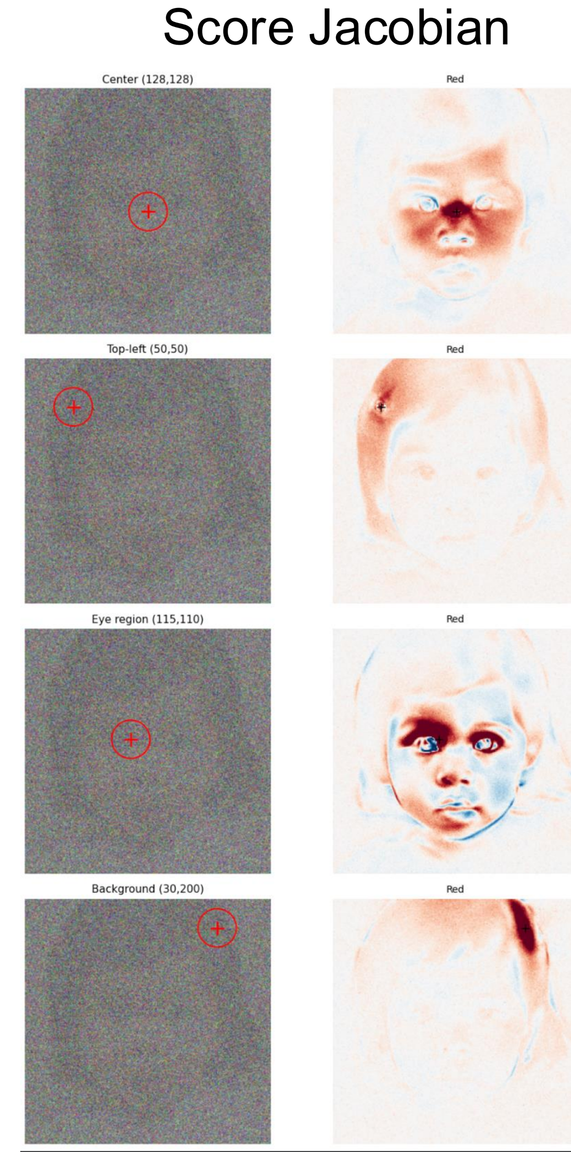
- A compressed view on the NN

Max timestep constraint: $\rho(J_f)$

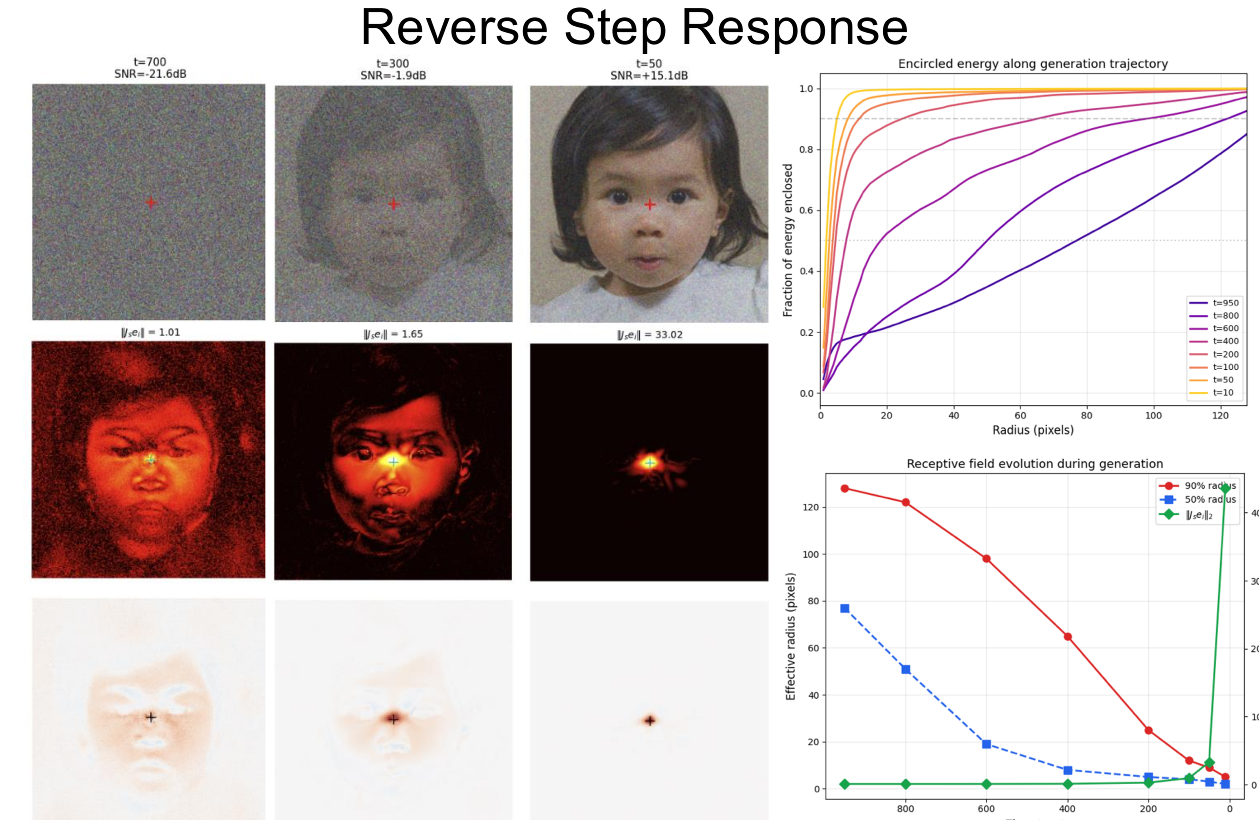
- Contrived connection to ODE stability

Experimental Results

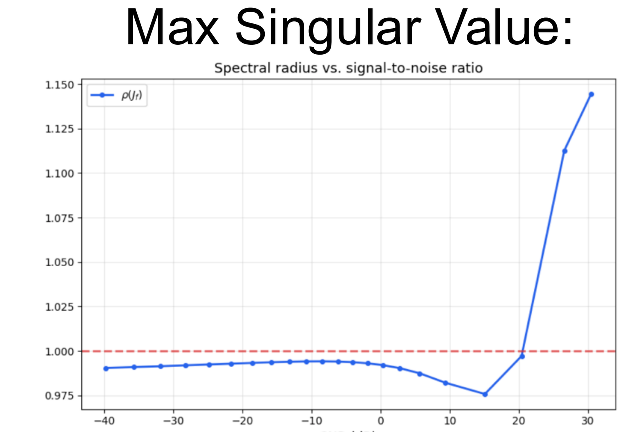
Score Jacobian



Reverse Step Response



Max Singular Value:

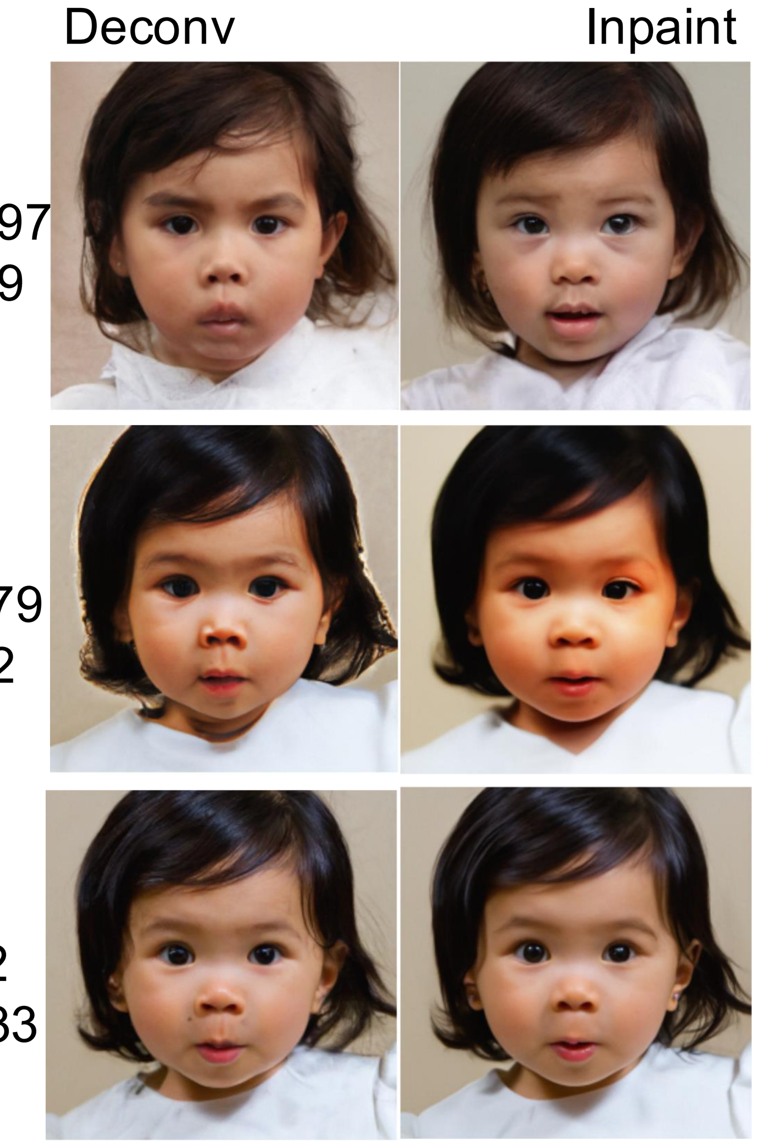


Deconv Inpaint

SDEdit
PSNR- 20.29/20.97
LPIPS- 0.22/0.19

ScoreALD
PSNR 22.38/22.79
LPIPS 0.14/0.12

DPS
PSNR 29.1/33.2
LPIPS 0.048/0.033



“Eigenfaces”: