Final Project Poster

● Ungraded

Student

Mike Zhao

View or edit group

Total Points

- / 1 pts

Question 1

Poster 1 pt

Inverse Imaging via Drifting Generative Models

Zixuan Zhao

Introduction

Problem: Inverse imaging problems (like super-resolution) are **ill-posed**. Generative models, like diffusion models, have been successful, but iterative inference is **expensive**.

Drifting Generative Models uses a drifting field to attract generated samples towards real data and repel them from other generated samples during training. This enables **one-step inference** with **stable training**.

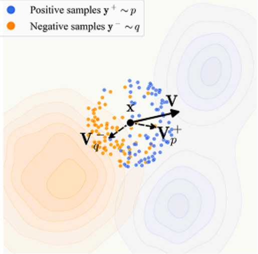
Related Work:

M. Deng, H. Li, T. Li, Y. Du, and K. He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.
Prafulla Dhariwal, Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arXiv preprint arXiv:2105.05233*, 2021
Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. *arXiv preprint arXiv:1809.00219*, 2018

Core Idea

Pushforward $f_\theta : \mathbb{R}^C \to \mathbb{R}^D$

$$\boldsymbol{\epsilon} \sim p_\epsilon$$
$$\mathbf{x} = f_\theta(\boldsymbol{\epsilon}) \sim q$$
$$\Delta \mathbf{x}_i = f_{i+1}(\boldsymbol{\epsilon}) - f_i(\boldsymbol{\epsilon})$$



Drifting Field $\mathbf{V}_{p,q}(\mathbf{x})$

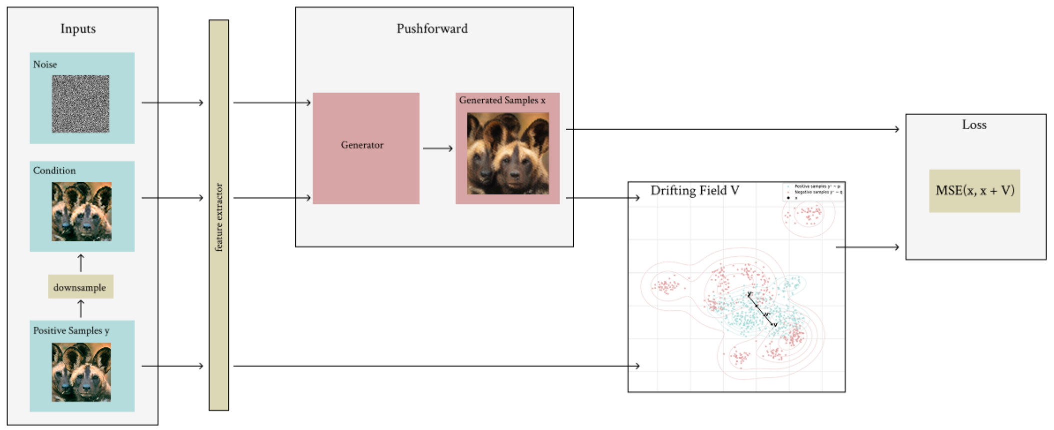
$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{V}_{p,q_i}(\mathbf{x}_i)$$
$$q = p \Rightarrow \mathbf{V}_{p,q} = \mathbf{0}$$
$$\mathbf{V}_p^+(\mathbf{x}) = \frac{1}{Z_p}\mathbb{E}_p[k(\mathbf{x}, \mathbf{y}^+)(\mathbf{y}^+ - \mathbf{x})] \qquad \mathbf{y}^+ \sim p_{\text{data}} \qquad \mathbf{y}^- \sim q$$
$$\mathbf{V}_q^-(\mathbf{x}) = \frac{1}{Z_q}\mathbb{E}_q[k(\mathbf{x}, \mathbf{y}^-)(\mathbf{y}^- - \mathbf{x})] \qquad Z_p, Z_q \text{ are normalization factors}$$
$$\mathbf{V}_{p,q}(\mathbf{x}) = \mathbf{V}_p^+(\mathbf{x}) - \mathbf{V}_q^-(\mathbf{x}) \qquad k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|/\tau)$$

Objective

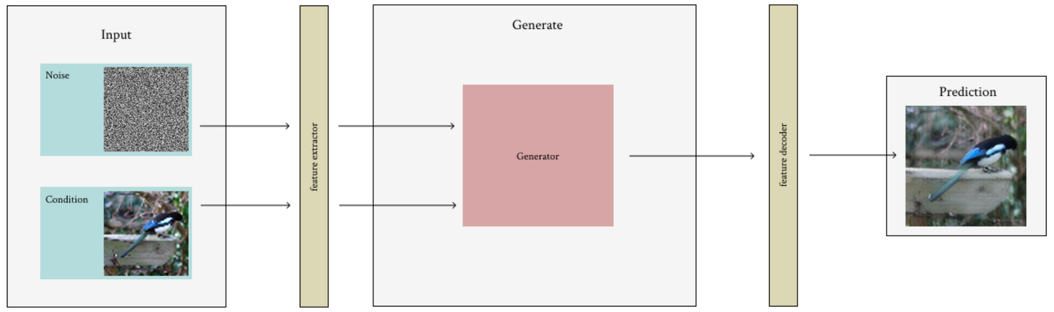
$$\mathcal{L} = \mathbb{E}_\epsilon\left[\|f_\theta(\boldsymbol{\epsilon}) - \text{stopgrad}(f_\theta(\boldsymbol{\epsilon}) + \mathbf{V}(f_\theta(\boldsymbol{\epsilon})))\|^2\right]$$
$$\mathcal{L} = \sum_j \mathbb{E}\left[\|\phi_j(\mathbf{x}) - \text{stopgrad}(\phi_j(\mathbf{x}) + \mathbf{V}(\phi_j(\mathbf{x})))\|^2\right]$$

Methodology

Drift Model Training

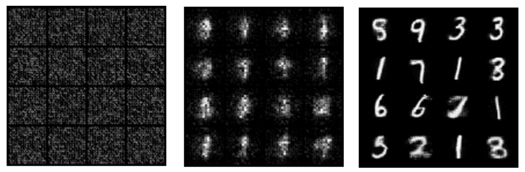


One-step Inference (super-resolution)

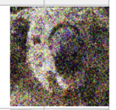


Pixel Space vs Latent Space

MNIST



Pixel vs Latent Space (10k Steps)

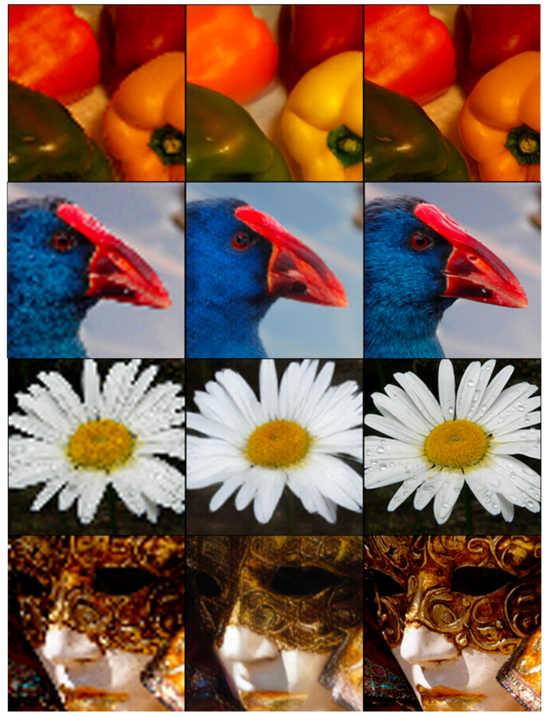
	PSNR	SSIM	FID
Pixel	15.98	0.4042	
DINOv3 + ViT-XL	18.10	0.4677	

Conclusion

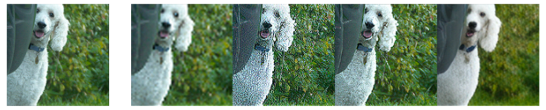
Takeaways:

Drifting models give near-diffusion image quality on inverse reconstruction (similar PSNR/SSIM, much better FID here) while being orders of magnitude faster at test

Results



Downsampled (4x) Generated (drift model) Ground Truth



Ground Truth Bicubic Upsampling ESRGAN Guided Diffusion Drifting

Evaluation

Test set = ~1000 images

Models	PSNR	SSIM	FID	Inference Time (ms per image)
Bicubic (baseline)	**24.55**	**0.7192**	80.11	**0.0042**
ESRGAN	16.20	0.3279	140.64	3.4
Guided Diffusion	20.15	0.5390	77.79	4284.8
Drifting Model	18.67	0.4894	**12.71**	1.8

time because inference is one-step.