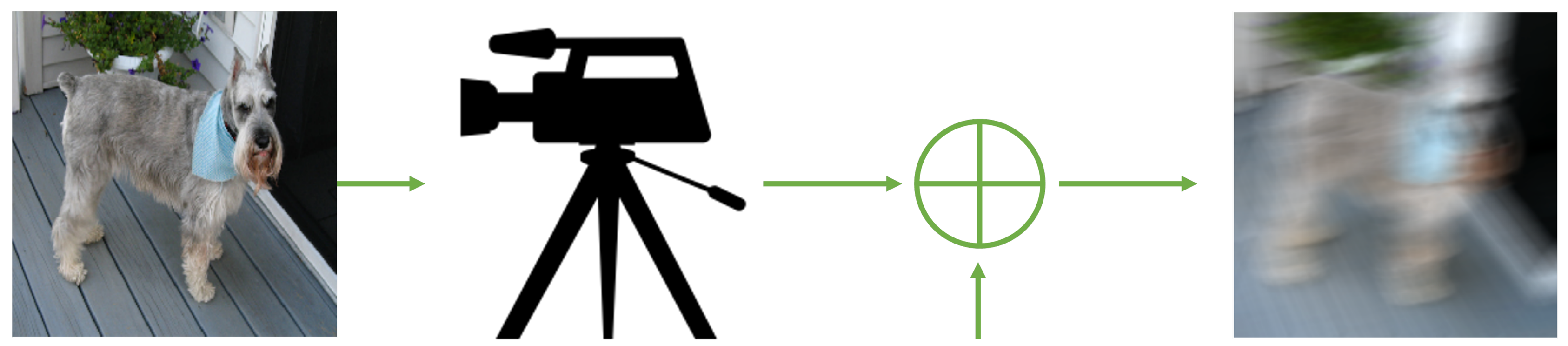
Solving Linear Inverse Problems Provably via Posterior Sampling using Latent Diffusion Models

Litu Rout Negin Raoof Giannis Daras Constantine Caramanis Alex Dimakis Sanjay Shakkottai
{litu.rout,neginmr,giannisdaras,constantine,sanjay.shakkottai}@utexas.edu, dimakis@austin.utexas.edu
The University of Texas at Austin

Introduction to Inverse Problems



Groundtruth Image $x_0 \sim p_{data}(X_0)$ | Imaging System A | Noise n | Measurements $y = A(x_0) + \sigma_y n$

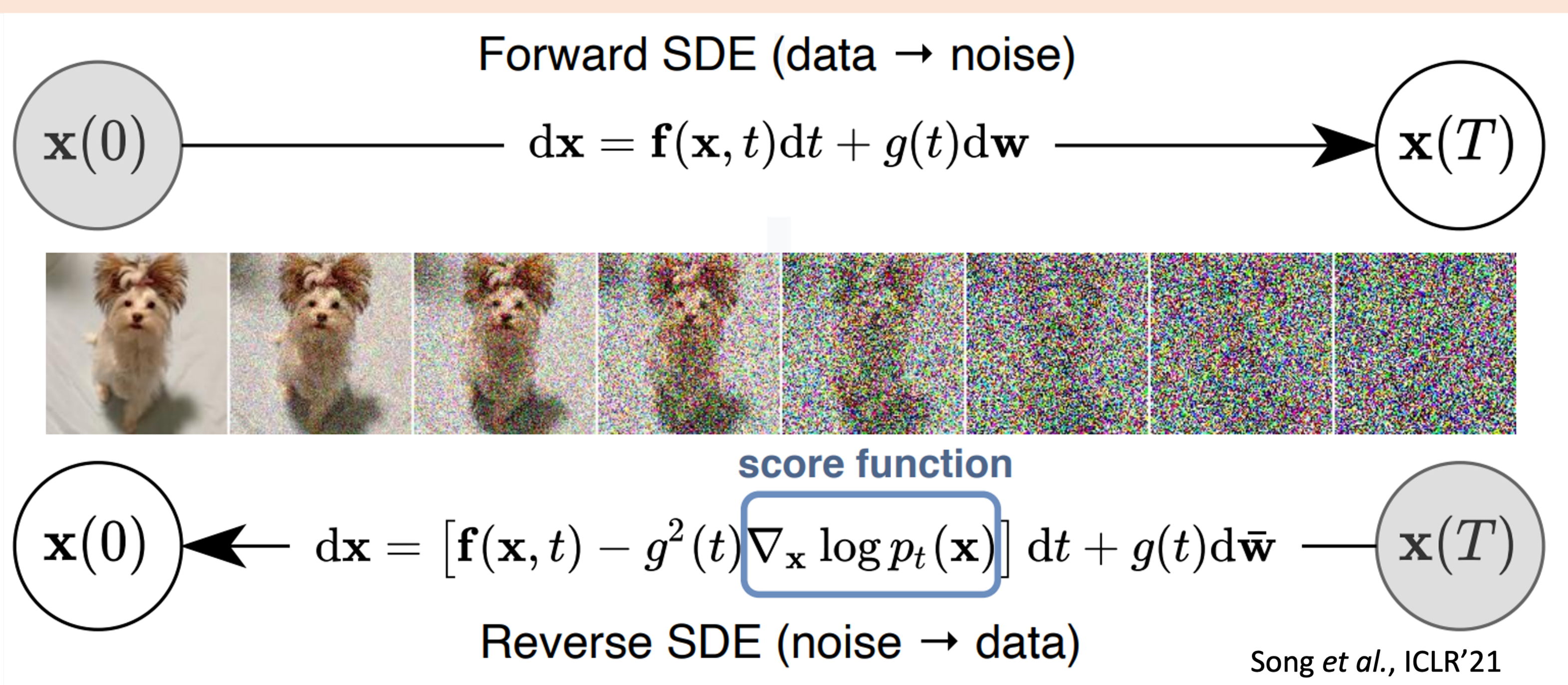
Problem: Reconstruct groundtruth image x_0 from noisy measurements y

Challenge: Problem is ill-posed, that is infinitely many solutions x_0 exist

Approach: Use prior knowledge $p(x_0)$ of how the image should look like

Background on Diffusion Models

Diffusion models have emerged as powerful priors for inverse problems!



Posterior Sampling using Diffusion Models

Problem: Sample $p_0(x_0|y)$ instead of $p(x_0)$

$$dX_t = (-X_t - 2\nabla \log p_t(X_t|y))\, dt + \sqrt{2}d\bar{W}_t, \qquad t = T, \cdots, 0$$

Unknown

$$dX_t = (-X_t - 2\nabla \log p_t(y|X_t) - 2\nabla \log p_t(X_t))\, dt + \sqrt{2}d\bar{W}_t \qquad \text{(Bayes rule)}$$

Unknown Known: $\nabla \log p_t(X_t) \approx s_\theta(X_t, t)$

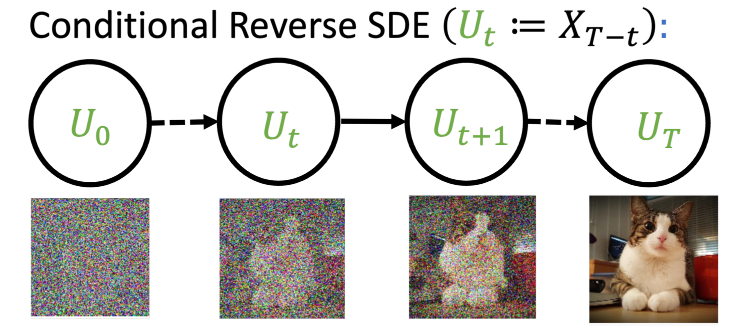
Approach: Sample $p_0(x_0|y)$ using $\nabla \log p(x_t)$ without having to re-train

1. a new score function $\nabla \log p_t(x_t|y)$ or
2. a noise-conditional measurement model $\nabla \log p_t(y|x_t)$

How well can we approximate $\nabla \log p_t(y|x_t)$?

Posterior Sampling using Diffusion Models

Conditional Reverse SDE ($U_t := X_{T-t}$):



At t, we need $p_{T-t}(y|U_t)$;
what we can compute instead is $p_{T-t}(y|U_T)$

Limitations of DPS:

- Hard to scale to high-resolution images
- Requires per-dataset generative prior models
- Hard to handle real-world images

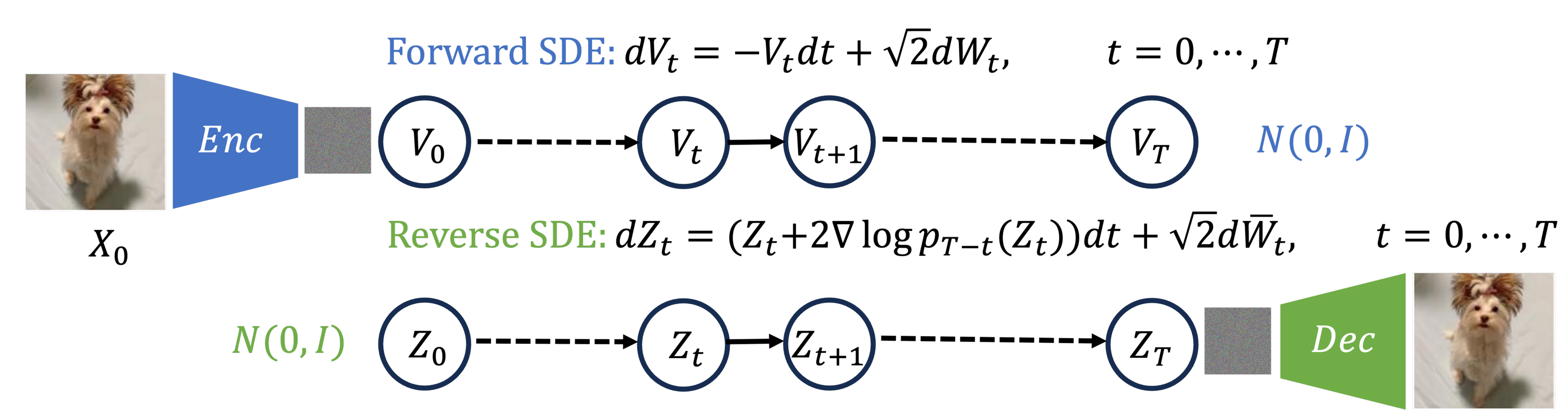
Our Approach: Diffuse in low-dimensional latent space using pretrained latent diffusion models, such as Stable Diffusion

DPS (Chung *et al.*, ICLR'23):

$$p_{T-t}(y|u_t) = E_{p_{T-t}(u_T|u_t)}[p_{T-t}(y|u_T)] \approx p_{T-t}\left(y|E_{p_{T-t}(u_T|u_t)}[u_T]\right)$$

can be computed using Tweedie's formula

Posterior Sampling using Latent Diffusion Models (Our Approach)



Posterior sampling using Latent-Space Diffusion Models (e.g. Stable Diffusion)

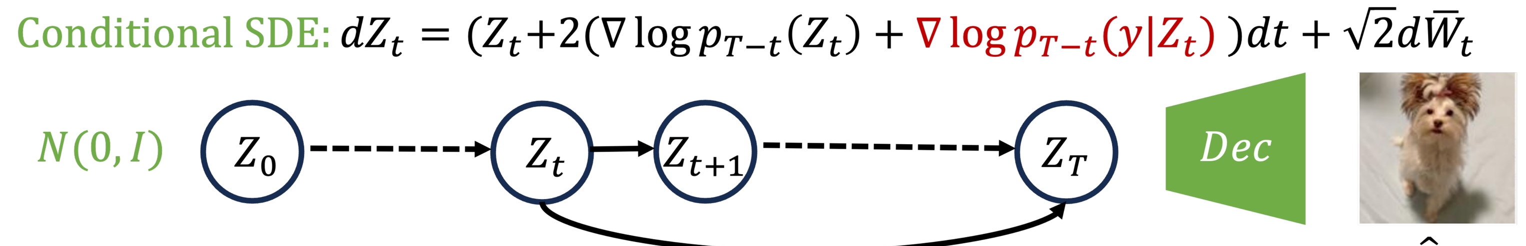
Stable Diffusion V-1.5
$V_t \in R^{64\times 64}$ and $X_t \in R^{512\times 512}$

$$dZ_t = (Z_t + 2(\nabla \log p_{T-t}(Z_t) + \nabla \log p_{T-t}(y|Z_t))\, dt + \sqrt{2}\, d\bar{W}_t$$

Problem: How well can we approximate $\nabla \log p_{T-t}(y|z_t)$?

Posterior Sampling using Latent Diffusion Models (Our Approach)

Conditional SDE: $dZ_t = (Z_t + 2(\nabla \log p_{T-t}(Z_t) + \nabla \log p_{T-t}(y|Z_t))dt + \sqrt{2}d\bar{W}_t$



At time t we need $p_{T-t}(y|Z_t)$; what we can compute instead is $p_{T-t}(y|Dec(Z_T))$

PSLD (Our approach): Denote by $\bar{Z}_T = E_{p_{T-t}(Z_T|z_t)}[Z_T]$. Then,

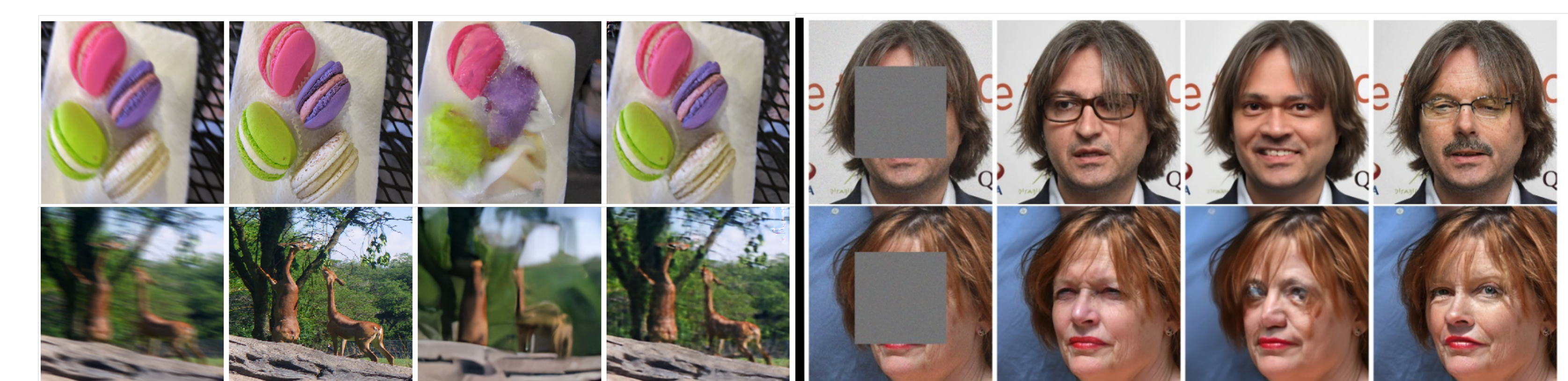
$$\nabla \log p_{T-t}(y|z_t) \approx \nabla \log p_{T-t}(y|Dec(\bar{Z}_T)) + \gamma_t \nabla \left|\left|\bar{Z}_T - Enc(A^T y + (I - A^T A)Dec(\bar{Z}_T))\right|\right|^2$$

- VAE is trained using $Dec(Enc(x_0)) = x_0$, $x_0 \sim p_{data}$
- Ideally, $\hat{x}_0 = Dec(z_T)$ is a natural image and $Enc(Dec(z_T)) = z_T$
- In practice, $Enc(Dec(z_T)) \neq z_T$, which causes instability and inconsistency
  - Stability: Look for z_T satisfying $Enc(Dec(z_T)) = z_T$
  - Consistency: Make sure that $A^T y + (I - A^T A)Dec(\bar{Z}_T)$ is a natural image

Experimental Results: Real-world Images

(a) Input (b) Groundtruth (c) Comm. Serv. 1 (d) Comm. Serv. 2 (e) PSLD (Ours)

Experimental Results: Motion Deblur (left) and Box Inpainting (right)



Input Groundtruth DPS PSLD (ours) | Input Groundtruth DPS PSLD (ours)

Method	Inpaint (random)		Inpaint (box)		SR (4×)		Gaussian Deblur	
	FID (↓)	LPIPS (↓)	FID (↓)	LPIPS (↓)	FID (↓)	LPIPS (↓)	FID (↓)	LPIPS (↓)
PSLD (Ours)	**21.34**	**0.096**	43.11	**0.167**	**34.28**	**0.201**	**41.53**	**0.221**
DPS [11]	33.48	0.212	**35.14**	0.216	39.35	0.214	44.05	0.257
DDRM [26]	69.71	0.587	42.93	0.204	62.15	0.294	74.92	0.332
MCG [13]	29.26	0.286	40.11	0.309	87.64	0.520	101.2	0.340
PnP-ADMM [6]	123.6	0.692	151.9	0.406	66.52	0.353	90.42	0.441
Score-SDE [47]	76.54	0.612	60.06	0.331	96.72	0.563	109.0	0.403
ADMM-TV	181.5	0.463	68.94	0.322	110.6	0.428	186.7	0.507

Method	Inpaint (random)	
	PSNR (↑)	SSIM (↑)
PSLD (Ours)	**30.31**	**0.851**
GML-DPS (Ours)	29.49	0.844
DPS [11]	25.23	**0.851**
DDRM [26]	9.19	0.319
MCG [13]	21.57	0.751
PnP-ADMM [6]	8.41	0.325
Score-SDE [47]	13.52	0.437
ADMM-TV	22.03	0.784

Experimental Results: Web Application

Input PSLD (Ours) Commercial