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

Speaker: Litu Rout

Based on joint work with: Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G. Dimakis, and Sanjay Shakkottai

The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
Cockrell School of Engineering

TEXAS
The University of Texas at Austin

Posterior Sampling

Problem Setup

Given measurements y and a measurement operator A, find a sample x that satisfies

$$y = Ax + n,$$

where $x \in R^d$, $A \in R^{k \times d}$, $y \in R^k$, $n \in N(0, \sigma_y^2 I_k)$.

Posterior Sampling & Optimization

For any $x \sim P$, let us denote by $P(y|x) := N(y; Ax, \sigma_y^2 I)$ the probability density of the measurement y given x. Given $y \sim P(y|x)$, the goal is to sample from $P(x|y)$.

$$x^* = \arg\max_x \log P(x|y) \propto \log\big(P(y|x)P(x)\big) = \log P(y|x) + \log P(x)$$

Posterior   Bayes' theorem   Likelihood   Prior

Stable Diffusion model has emerged as a powerful new prior for sampling $P(x|y)$.

Challenges for Posterior Sampling

Posterior sampling present two unique challenges:

i. **Challenge 1:** Inexact score function

$$dx = \left(f(x,t) - g^2(t)\nabla_{x_t} \log P_t(x_t|y)\right) dt + g(t)\, dw$$

- Require $\nabla_{x_t} \log P_t(x_t|y)$, have access to $\nabla_{x_t} \log P_t(x_t) \approx S_\theta(x_t, t)$

Get from Diffusion Model ☺

i. **Challenge 2:** Likelihood approximation

- Posterior sampling requires samples from $P(x|y)$:

$$dx = \left(f(x,t) - g^2(t)\left(\nabla_{x_t} \log P_t(x_t) + \nabla_{x_t} \log P_t(y|x_t) \right)\right) dt + g(t)\, dw$$

Question: How well can we approximate $\nabla_{x_t} \log P_t(y|x_t)$?

Method: Diffusion in Pixel-Space

Posterior sampling using Pixel-Space Diffusion Models

- Posterior sampling requires samples from $P(x|y)$:

$$dx = \left(f(x,t) - g^2(t)(\nabla_{x_t} log\, P_t(x_t) + \nabla_{x_t} log\, P_t(y|x_t)\right) dt + g(t)\, dw$$

Question: How well can we approximate $\nabla_{x_t} log\, P_t(y|x_t)$?

DPS approximation [Chung *et al.* ICLR'2023]:

$$P_t(y|x_t) \approx P(y|\widehat{x_0} = E[x_0|x_t]) = N(y; \mu = A\widehat{x_0}, \sigma = \sigma_y I)$$

where $\widehat{x_0} = \frac{1}{\sqrt{\overline{\alpha_t}}}(x_t + (1 - \overline{\alpha_t})S_\theta(x_t, t))$

Tweedie's formula

Get from Diffusion Model ☺

Challenge iii. Computationally very expensive. Requires gradients to be computed in the pixel space. Hard to scale to higher resolution images.

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

- Posterior sampling requires samples from $P(x|y)$:

$$dz = \left(f(z,t) - g^2(t)(\nabla_{z_t} \log P_t(z_t) + \nabla_{z_t} \log P_t(y|z_t)\right) dt + g(t)\, dw$$

Question: How well can we approximate $\nabla_{z_t} \log P_t(y|z_t)$?

Latent-DPS approximation [RRDCDS’2023]:

$$P_t(y|z_t) \approx P(y|Dec(\hat{z}_0)) = N(y; \mu = ADec(\hat{z}_0), \sigma = \sigma_y^2 I)$$

where $\hat{z}_0 = E[z_0|z_t] = \frac{1}{\sqrt{\overline{\alpha_t}}}(z_t + (1 - \overline{\alpha_t})\, \widehat{S_\theta}(z_t, t))$

Tweedie’s formula

Get from Stable Diffusion ☺

Stable Diffusion V-1.5

$z_t \in R^{64\times64}$ and $x_t \in R^{512\times512}$

Challenge iv. Many-to-one mapping of encoder. Training is unstable. Does not converge to the true underlying sample.

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

- Posterior sampling requires samples from $P(x|y)$:

$$dz = \left(f(z,t) - g^2(t)(\nabla_{z_t} log\, P_t(z_t) + \nabla_{z_t} log\, P_t(y|z_t)\right) dt + g(t)\, dw$$

Question: How well can we approximate $\nabla_{z_t} log\, P_t(y|z_t)$?

GML-DPS approximation [RRDCDS'2023]:

Look for a fixed point of the VAE

$$\nabla_{z_t} \log P(y|z_t) = \nabla_{z_t} P(y|x_0 = Dec(E[z_0|z_t])) + \gamma_t \nabla_{z_t} ||E[z_0|z_t] - Enc(Dec(E[z_0|z_t]))||$$

where $\widehat{z_0} = E[z_0|z_t] = \frac{1}{\sqrt{\overline{\alpha_t}}}(z_t + (1 - \overline{\alpha_t})\, \widehat{S_\theta}(z_t, t))$

Tweedie's formula

Get from Stable Diffusion ☺

Stable Diffusion V-1.5

$z_t \in R^{64\times64}$ and
$x_t \in R^{512\times512}$

Challenge v. Many potential solutions exist. Requires a specific choice of step size γ_t, see Theorem 3.7 [RRDCDS'2023)]

Method: Diffusion in Latent-Space

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

- Posterior sampling requires samples from $P(x|y)$:

$$dz = \left(f(z,t) - g^2(t)\left(\nabla_{z_t} log\, P_t(z_t) + \nabla_{z_t} log\, P_t(y|z_t)\right)\right) dt + g(t)\, dw$$

Question: How well can we approximate $\nabla_{z_t} log\, P_t(y|z_t)$?

Look for the fixed point of the VAE using gluing objective

PSLD approximation [RRDCDS'2023]:

$$\nabla_{z_t} \log P(y|z_t) = \nabla_{z_t} P(y|x_0 = Dec(E[z_0|z_t])) + \gamma_t \nabla_{z_t} ||E[z_0|z_t] - Enc(A^T y + (I - A^T A) Dec(E[z_0|z_t]))||$$

where $\hat{z_0} = E[z_0|z_t] = \frac{1}{\sqrt{\overline{\alpha_t}}}\left(z_t + (1 - \overline{\alpha_t})\,\widehat{S_\theta}(z_t, t)\right)$

Tweedie's formula

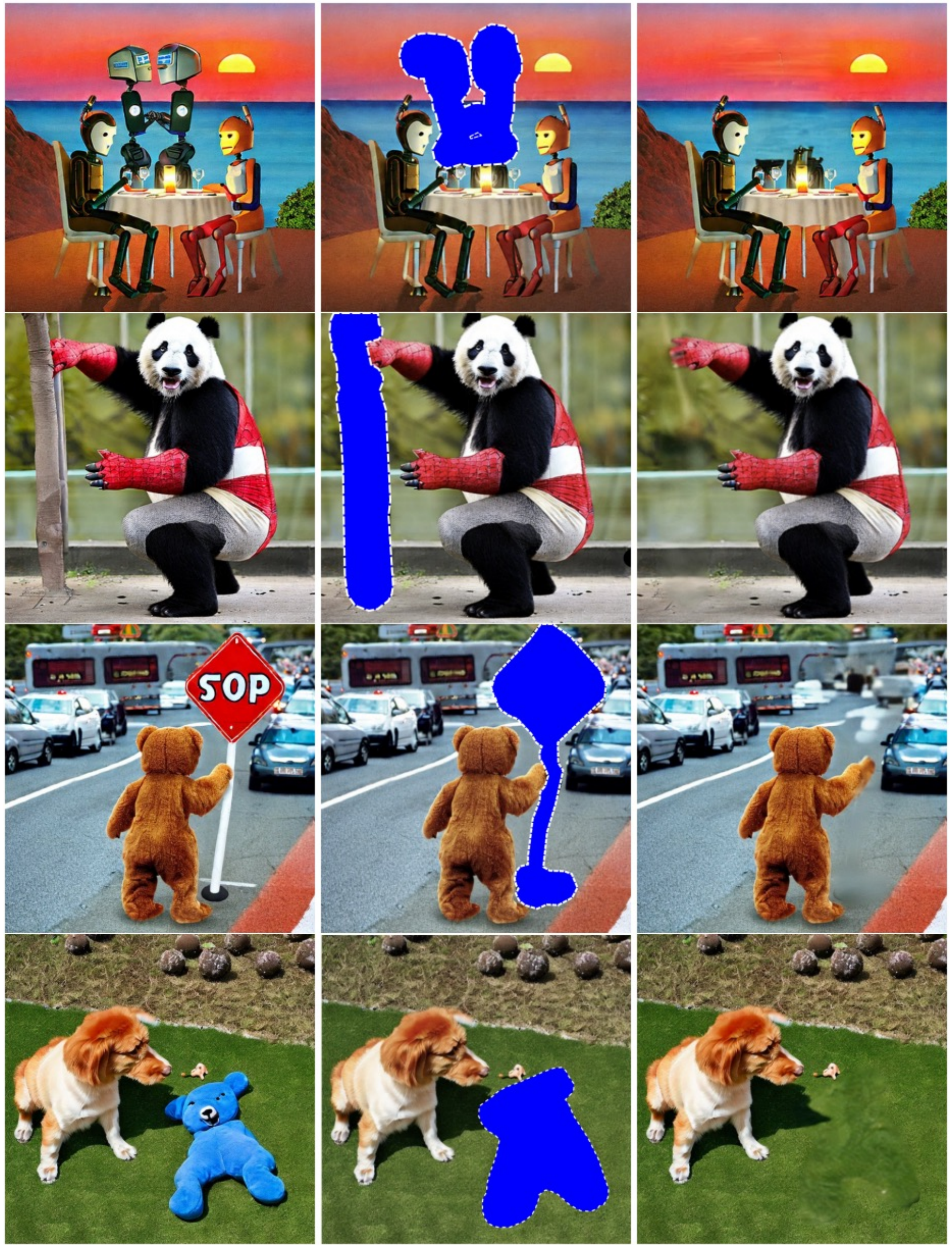
Get from Stable Diffusion ☺

Stable Diffusion V-1.5

$z_t \in R^{64\times64}$ and $x_t \in R^{512\times512}$

Solution: Gluing objective converges to unique solution if it exists. Works with any choice of step size γ_t, see Theorem 3.8 [RRDCDS'2023].

Experimental Results: Overall pipeline of our proposed framework from left to right



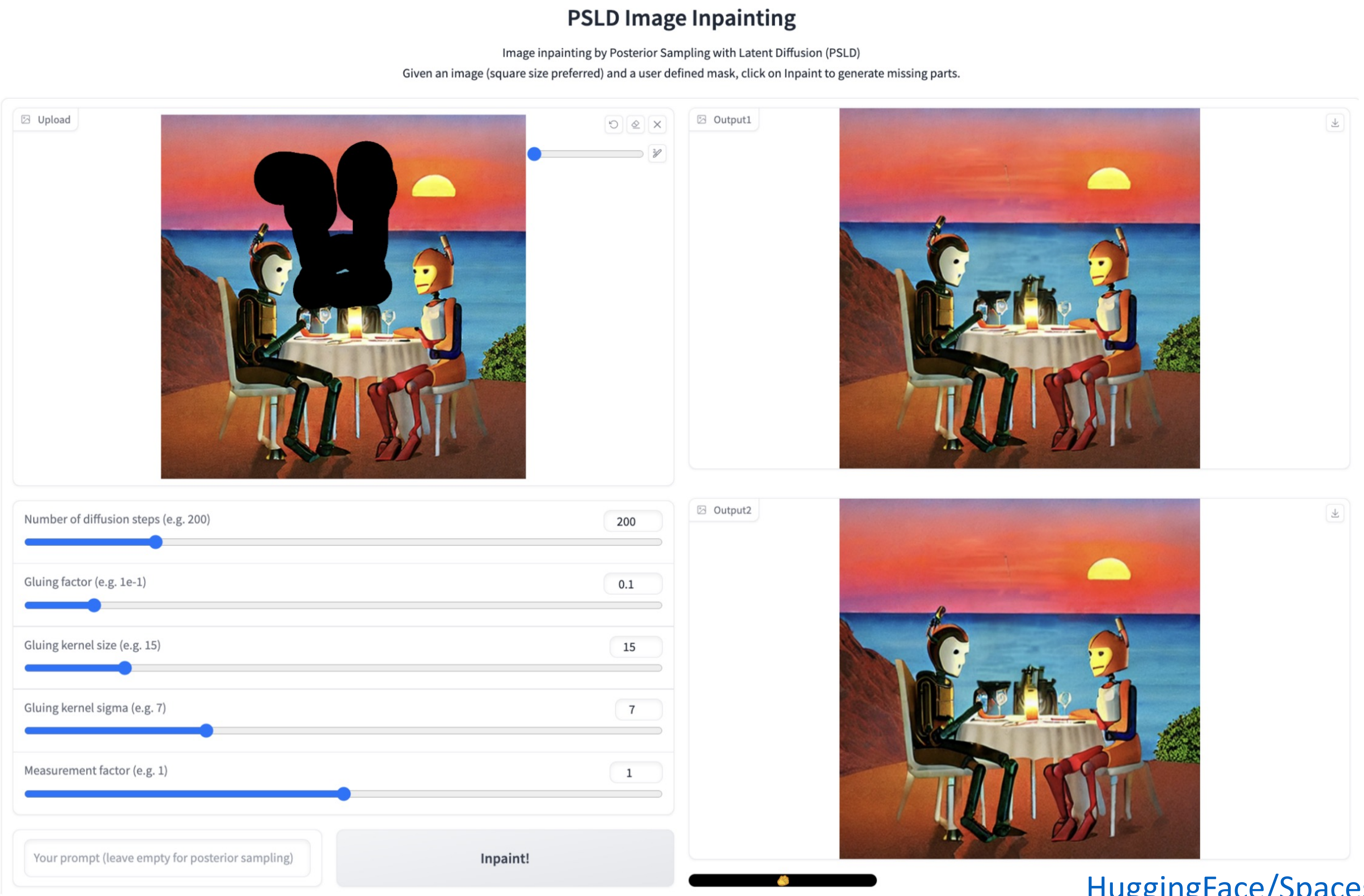
Experimental Results: Comparison with commercial inpainting services that use Stable Diffusion

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

Quantitative results on FFHQ 256x256 using Stable Diffusion V-1.5

	Inpaint (random)		Inpaint (box)		SR (4×)		Gaussian Deblur	
Method	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	<u>0.212</u>	**35.14**	0.216	<u>39.35</u>	<u>0.214</u>	<u>44.05</u>	<u>0.257</u>
DDRM [26]	69.71	0.587	42.93	<u>0.204</u>	62.15	0.294	74.92	0.332
MCG [13]	<u>29.26</u>	0.286	<u>40.11</u>	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

Experimental Results: Web application for user defined masks



HuggingFace/Spaces

Thanks for listening!

Any questions can be sent to:
litu.rout@utexas.edu

Solving Linear Inverse Problems Provably via Posterior Sampling with Latent Diffusion Models
arXiv:2307.00619
OpenReview
Code Demo