



On Solving Inverse Problems using Latent Diffusion

Speaker: Sanjay Shakkottai

Based on joint work with: Litu Rout, Negin Raoof, Giannis Daras, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Alexandros G. Dimakis and Wen-Sheng Chu

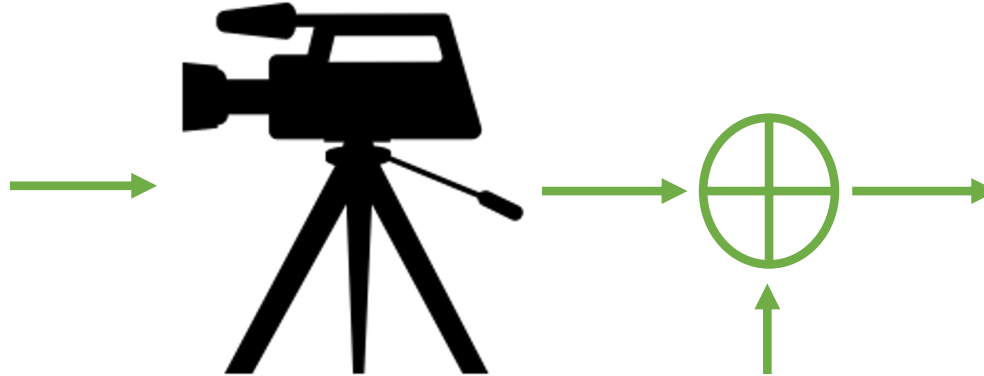
The University of Texas at Austin, Google Research, Google DeepMind



General Inverse Problems



Ground truth Image



Imaging System

Noise



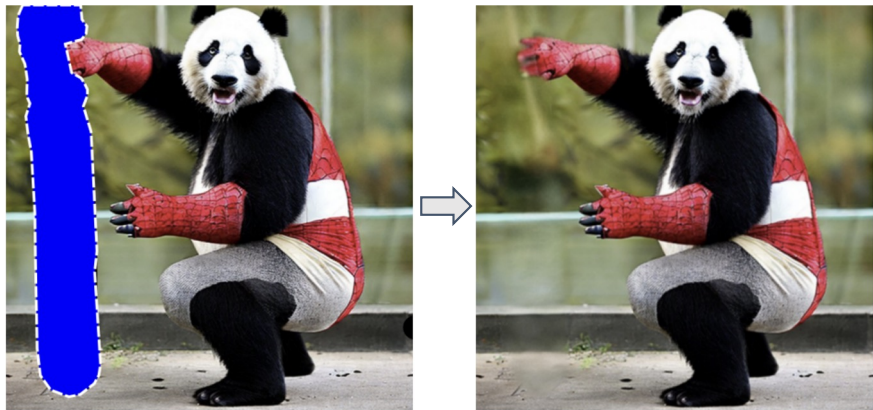
Measurements

Problem: Reconstruct ground truth image from noisy measurements

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

Approach: Use **prior** knowledge of how the image should look like

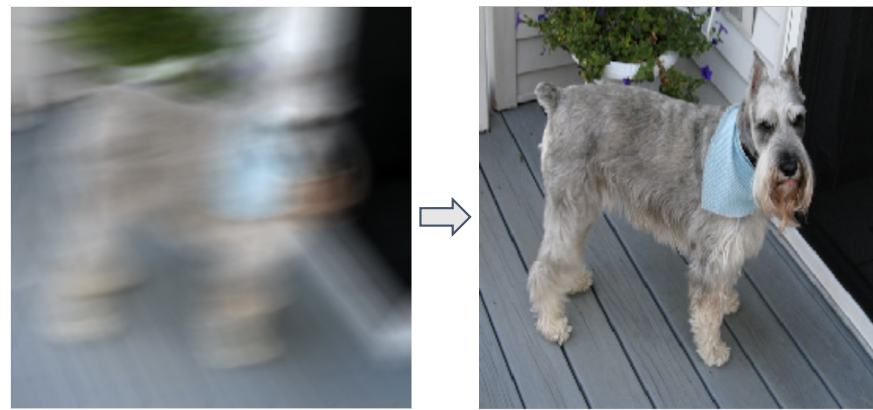
General Inverse Problems



Free-form inpainting



Super-resolution (4X)

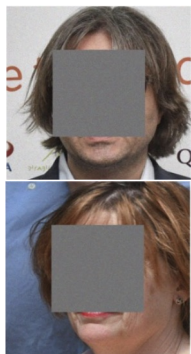
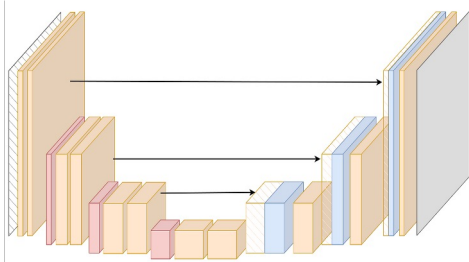


Motion Deblur

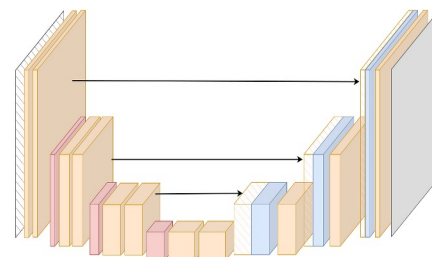
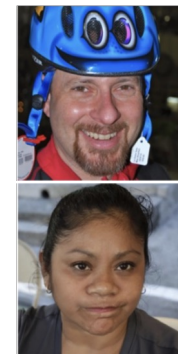


Gaussian Deblur

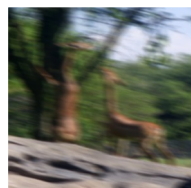
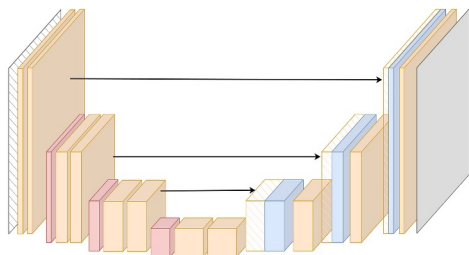
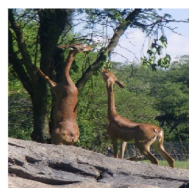
Inverse Problem Solvers: Supervised Methods



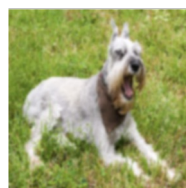
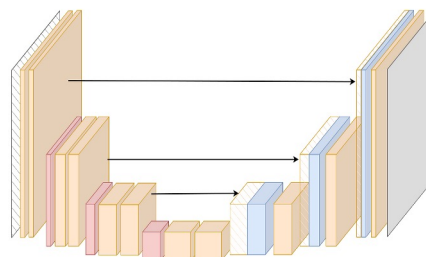
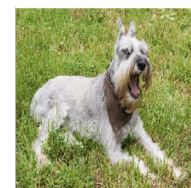
Box inpainting: Model 1



Random inpainting: Model 2

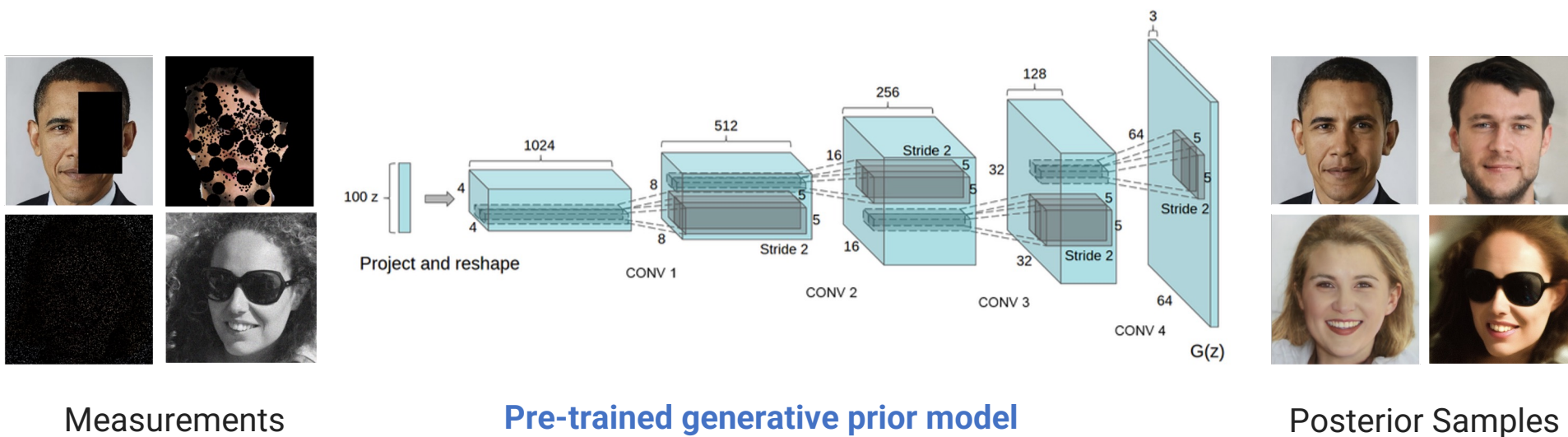


Motion Deblur: Model 3



Gaussian Deblur: Model 4

Inverse Problem Solvers: Unsupervised Methods



Solve **many inverse problems** using the same **pre-trained generative prior model**

Inverse Problem Solvers: Unsupervised Methods

- Wide availability of generative foundation models, e.g., Stable Diffusion, Imagen, DALL-E
- No expensive re-training needed for different inverse tasks, can be achieved with less annotations
- Leverage data and compute investment of large-scale foundation models
- Sample from posterior distribution instead of regressing to the mean

Examples of Generative Models:

- Variational AutoEncoders (Kingma & Welling' 2013)
- Generative Adversarial Networks (Goodfellow *et al.*' 2014)
- Deep Unsupervised Learning using Diffusion (Sohl-Dickstein *et al.*' 2015)
- Score-based Generative Models (Song & Ermon' 2019)
- Diffusion Probabilistic Models (Ho *et al.*' 2020)
- Score-based Generative Models through SDEs (Song *et al.*' 2021)

Diffusion models have recently emerged as powerful generative priors for solving general inverse problems!

Background on Diffusion Models

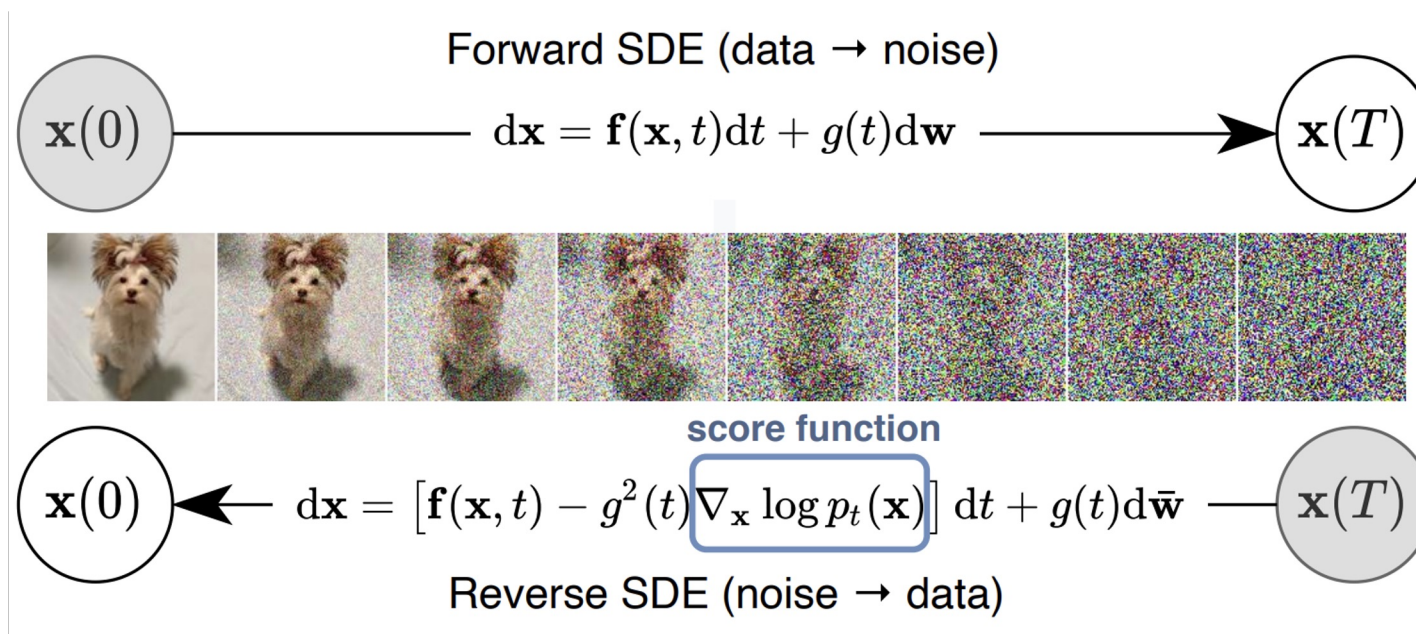


Image Credit: Song et al., <https://arxiv.org/pdf/2011.13456.pdf>, ICLR'21

Background on Diffusion Models

First, add noise to samples from $p_0(x_0) := p_{data}(x)$ using forward *Ito* SDE:

$$dX_t = -X_t dt + \sqrt{2}dW_t, \quad t = 0, \dots, T$$

Second, sample $p_0(x_0)$ using reverse *Ito* SDE:

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

Third, learn the score-function $\nabla \log p_t(x_t)$ at all noise levels t using Denoising Score Matching (DSM) [Tweedie'1950, Hyvarinen'2005, Vincent'2011]

$$dX_t = (-X_t - 2s_\theta(X_t, t)) dt + \sqrt{2}d\bar{W}_t, \quad t = T, \dots, 0$$

Background on Diffusion Models (Tweedie's Formula)

Proposition (Tweedie' 1950, Efron'2011)

For $X_0 \sim p_0(\cdot)$, $\epsilon \sim N(0, I)$, suppose $X_t = X_0 + \sigma_t \epsilon$. Then, the MMSE estimator of X_0 given X_t is given by $E_{X_0 \sim p_0 | t(X_0 | X_t)}[X_0] = X_t + \sigma_t^2 \nabla \log p_t(X_t)$

Theorem (Tweedie' 1950, Hyvarinen'2005, Vincent'2011)

$$J_1(\theta) = E_{X_t} \left[\|S_\theta(X_t, t) - \nabla \log p_t(X_t)\|_2^2 \right] \quad \text{and} \quad J_2(\theta) = E_{X_0} E_{X_t | X_0} \left[\|S_\theta(X_t, t) - \nabla \log p_t(X_t | X_0)\|_2^2 \right]$$

Then, the following holds: $J_2(\theta) = J_1(\theta) + \text{constant}$.

Key Takeaways

- Need to solve $\theta_1^* = \arg \min_{\theta} J_1(\theta)$
- Can not be solved because don't know $\nabla_{x_t} \log p_t(x_t)$
- Instead, solve $\theta_2^* = \arg \min_{\theta} J_2(\theta)$
- **Fact:** The constant in the above theorem is independent of θ , so $\theta_2^* = \theta_1^*$.
 $\theta_2^* = \arg \min_{\theta} J_2(\theta) = \arg \min_{\theta} J_1(\theta) + \text{constant} = \arg \min_{\theta} J_1(\theta) = \theta_1^*$

Posterior Sampling using Diffusion Models

Focus on three posterior sampling methods:

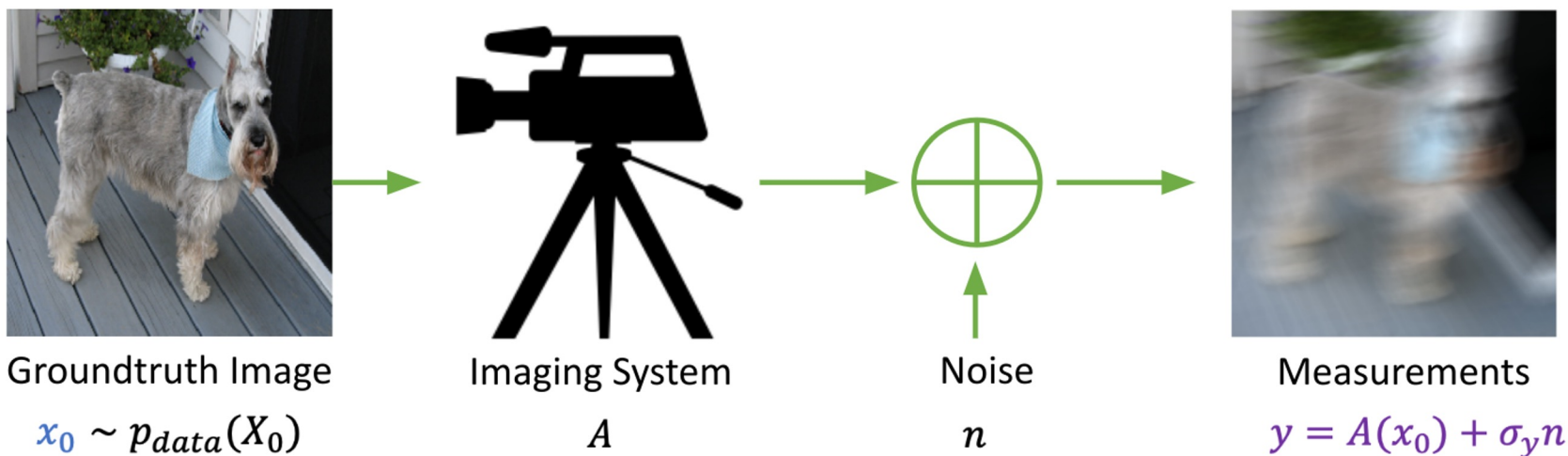
- DPS: Diffusion Posterior Sampling (Chung et al., ICLR'2023)
 - First-order Tweedie Approximation in **pixel space**
- PSD: Posterior Sampling using Latent Diffusion (Rout et al., NeurIPS'2023)
 - First-order Tweedie Approximation in **latent space**
- STSL: Second-order Tweedie from Surrogate Loss (Rout et al., CVPR'2024)
 - **Second-order** Approximation Tweedie in latent space

Background on
Posterior Sampling
using Diffusions

“**Diffusion Posterior Sampling for General Noisy Inverse Problems**”, Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, Jong Chul Ye, ICLR 2023

Background on Diffusion Posterior Sampling (DPS)

(Prior Work by Chung et al ICLR 2023)



Problem: Reconstruct ground truth 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 Posterior Sampling (DPS)

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

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

↓
Unknown

Bayes rule: $\log p_t(x_t|y) = \log p_t(y|x_t) + \log p_t(x_t) - \log p_t(y)$

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

↓
Unknown

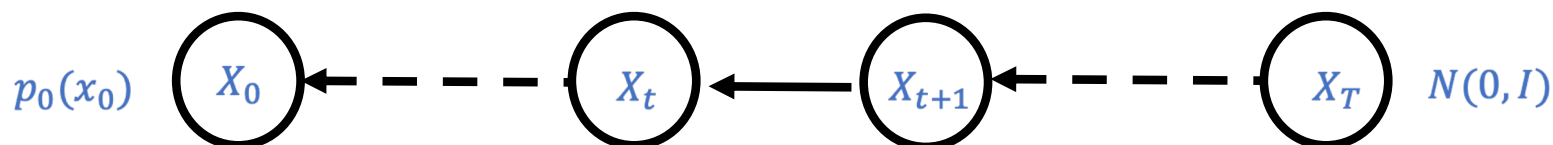
↓
Known: $\nabla \log p_t(X_t) \approx s_\theta(X_t, t)$

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

Background on Diffusion Posterior Sampling (DPS)

Reverse SDE ($t = T, T - 1, \dots, 0$):

$$dX_t = (-X_t - 2\nabla \log p_t(y|X_t) - 2\nabla \log p_t(X_t)) dt + \sqrt{2}dW_t$$

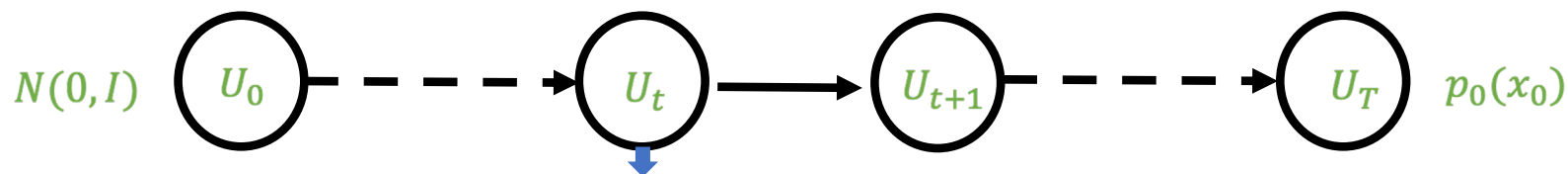


For convenience, denote random variables of the reverse process by $U_t := X_{T-t}$.

Reverse SDE ($t = 0, \dots, T - 1, T$):

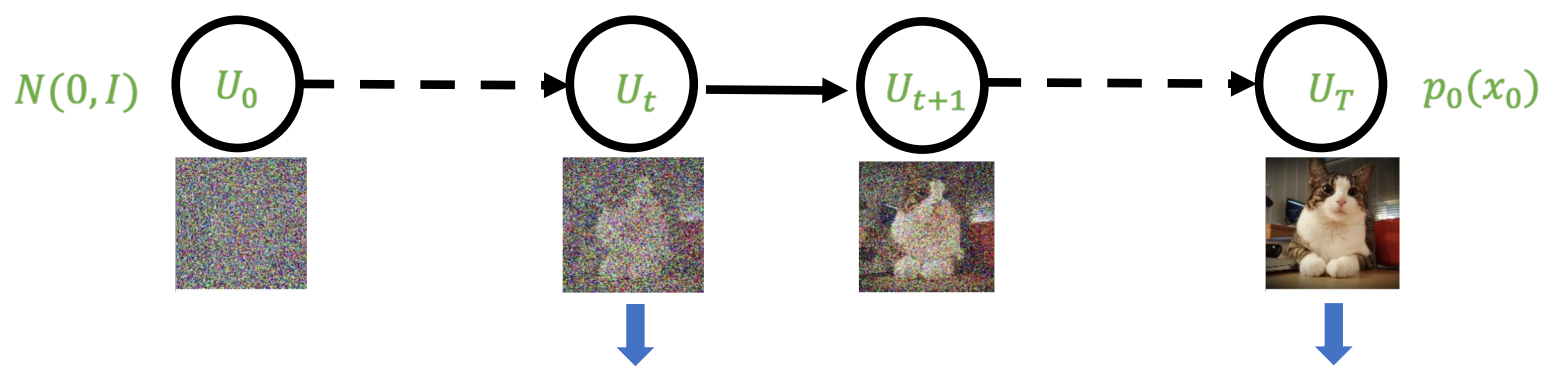
$$dU_t = (U_t + 2\nabla \log p_{T-t}(y|U_t) + 2\nabla \log p_{T-t}(U_t)) dt + \sqrt{2}d\bar{W}_t$$

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



At time t we need $p_{T-t}(y|U_t)$; what we can compute instead is $p_0(y|U_T)$

Background on Diffusion Posterior Sampling (DPS)



At time t we need $p_{T-t}(y|U_t)$; what we can compute instead is $p_0(y|U_T)$

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

$$p_{T-t}(y|u_t) = E_{U_T \sim p_{T-t}(U_T|u_t)}[p_0(y|U_T)] \approx p_0\left(y|E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T]\right)$$

Interpretation: Suppose $g(\cdot) = p_0(y|\cdot)$. Then, $E[g(\cdot)] \approx g(E[\cdot])$. We can compute $E[\cdot]$ using Tweedie's formula.

Limitations

- Hard to scale to higher resolution images
 - Gradients computed in pixel space
- Per-dataset generative prior model
 - FFHQ (human faces) and ImageNet
- Difficulty in real-world deployment
 - Images from the web, OOD samples

Our Approach: Diffuse in lower-dimensional latent space using pre-trained latent diffusion models, such as Stable Diffusion

Can we use a general prior, e.g. Stable Diffusion as opposed to specific priors used in DPS?

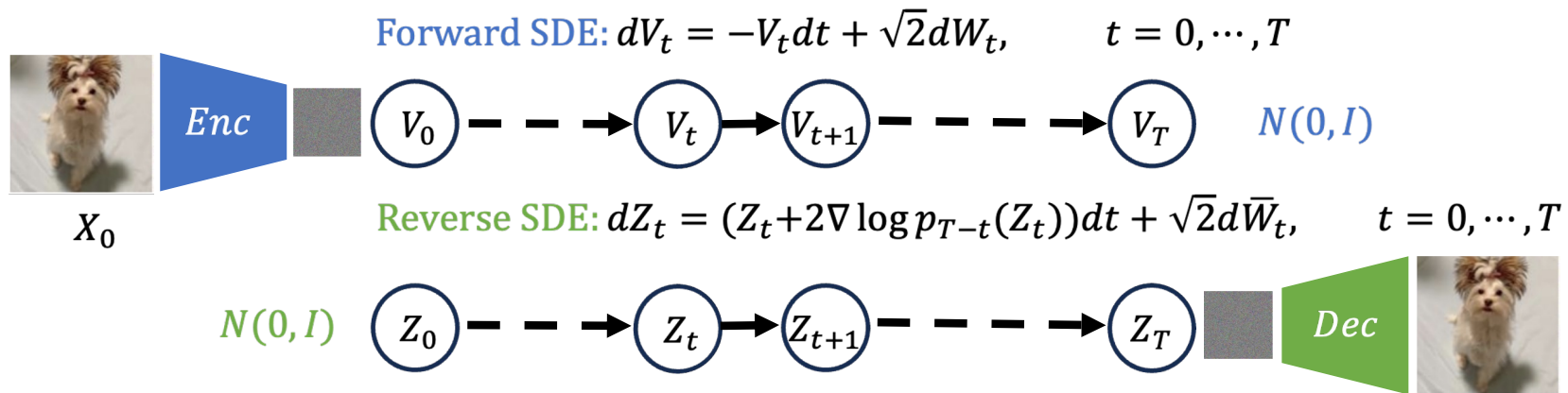
Latent Posterior Sampling using Diffusion Models

Focus on three posterior sampling methods:

- DPS: Diffusion Posterior Sampling (Chung et al., ICLR'2023)
 - First-order Tweedie in **pixel space**
- PS�D: Posterior Sampling using Latent Diffusion (Rout et al., NeurIPS'2023)
 - First-order Tweedie in **latent space**
- STSL: Second-order Tweedie from Surrogate Loss (Rout et al., CVPR'2024)
 - **Second-order** Tweedie in latent space

“**Solving Linear Inverse Problems Provably using Posterior Sampling with Latent Diffusion Models**”, Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G. Dimakis, and Sanjay Shakkottai, *NeurIPS 2023*

PSLD: Posterior Sampling using Latent Diffusion



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

Stable Diffusion V-1.5

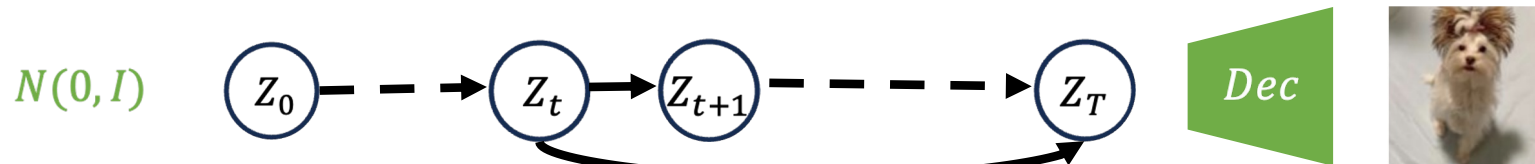
$V_t \in \mathbb{R}^{64 \times 64}$ and
 $X_t \in \mathbb{R}^{512 \times 512}$

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

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

PSLD: Posterior Sampling using Latent Diffusion

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 $p_0(y|Dec(Z_T))$

LDPS (Rout et al., NeurIPS'23):

$$p_{T-t}(y|Z_t) = E_{Z_T \sim p_{T-t}(Z_T|Z_t)} [p_0(y|Dec(Z_T))]$$

if Encoder and Decoder are one-to-one mappings

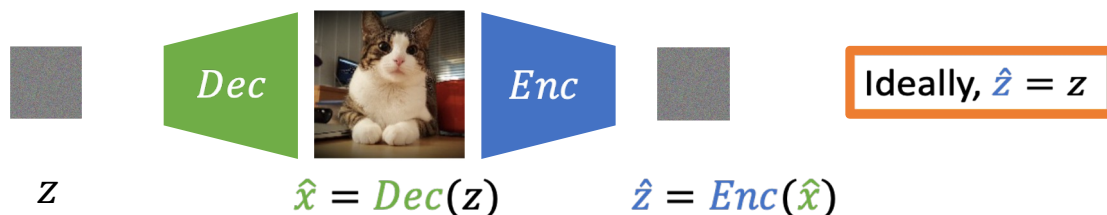
In practice, Encoder and Decoder are NOT one-to-one mappings

PSLD: Posterior Sampling using Latent Diffusion

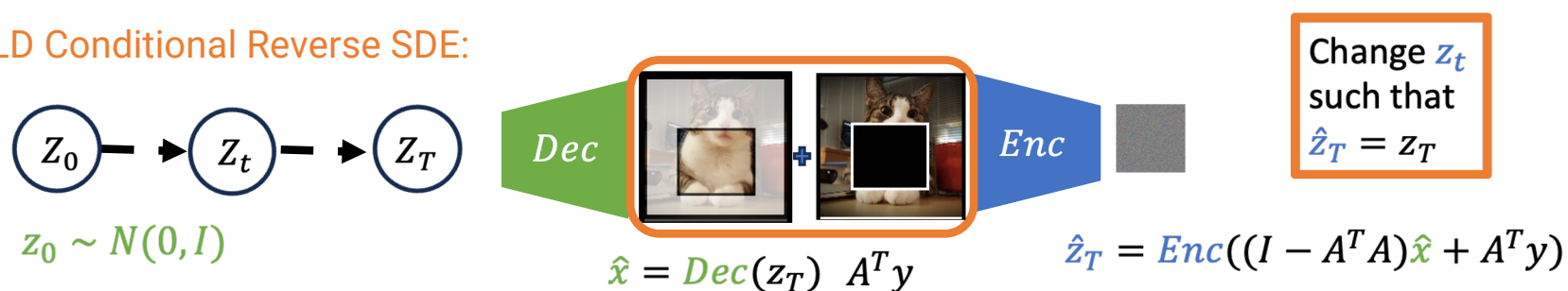
Sampling from LDPS is unstable

- Latent space of the VAE is a compressed subspace
- Encoder maps many high-dimensional pixel grids into the same low-dimensional latent

Inference:

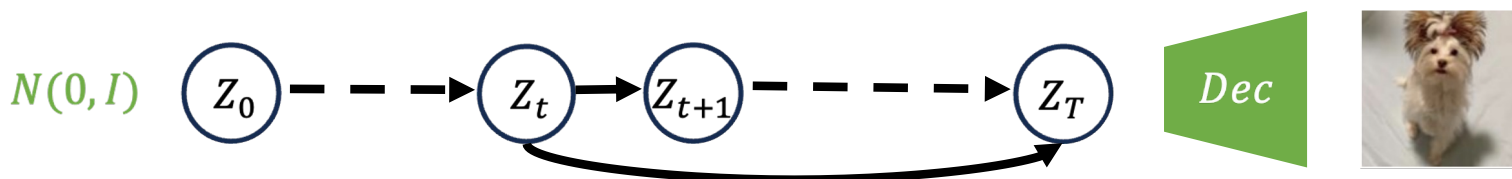


PSLD Conditional Reverse SDE:



PSLD: Posterior Sampling using Latent Diffusion

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 $p_0(y|Dec(Z_T))$

PSLD (Rout et al.,NeurIPS'23):

$$\nabla \log p_{T-t}(y|Z_t) \approx \nabla \log p_0(y|Dec(\bar{Z}_T)) + \gamma_t \nabla \left\| \bar{Z}_T - Enc(A^T y + (I - A^T A)Dec(\bar{Z}_T)) \right\|^2$$

where $\bar{Z}_T = E_{Z_T \sim p_{T-t}(Z_T|Z_t)}[Z_T]$

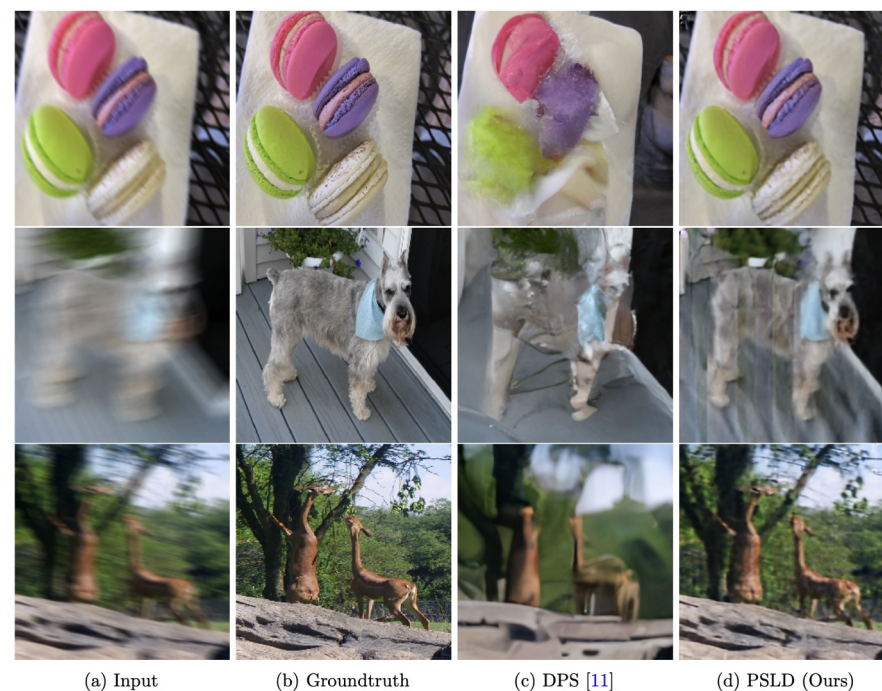
Gluings Objective:

- Look for a unique fixed point of the Encoder-Decoder of Stable Diffusion
- Works with **any** choice of step size γ_t for perfect recovery in a linear model setting

Experimental Results: Box inpainting and Motion Deblur



- Scalable to higher resolution images
 - Gradients computed in latent space
- Same generative prior model, Stable Diffusion V1.5
 - FFHQ (human faces) and ImageNet



- Convenient for real-world deployment
 - Images from the web, OOD samples

Experimental Results: Hugging Face Demo

PSLD Image Inpainting

Image inpainting by Posterior Sampling with Latent Diffusion (PSLD)
Given an image (square size preferred) and a user defined mask, click on Inpaint to generate missing parts.

Upload



Output1



Output2



Commercial Services



Number of diffusion steps (e.g. 200)

200

Gluing factor (e.g. 1e-1)

0.1

Gluing kernel size (e.g. 15)

15

Gluing kernel sigma (e.g. 7)

7

Measurement factor (e.g. 1)

1

Your prompt (leave empty for posterior sampling)

Inpaint!

Image Inpainting by [PSLD](#) - Generative Foundation Model by [RunwayML](#) - Gradio Demo by [Hugging Face](#)

PSLD
(Ours)

Commercial
Services

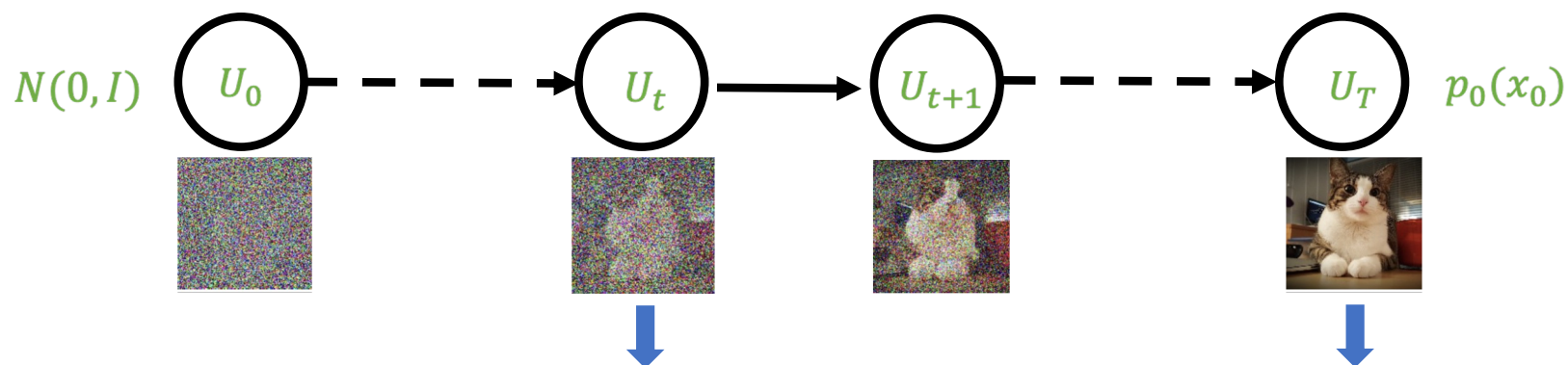
Second-order Latent Posterior Sampling

Focus on three posterior sampling methods:

- DPS: Diffusion Posterior Sampling (Chung et al., ICLR'2023)
 - First-order Tweedie in **pixel space**
- PSD: Posterior Sampling using Latent Diffusion (Rout et al., NeurIPS'2023)
 - First-order Tweedie in **latent space**
- STSL: Second-order Tweedie from Surrogate Loss (Rout et al., CVPR'2024)
 - **Second-order** Tweedie in latent space

“**Beyond First-Order Tweedie: Solving Inverse Problems using Latent Diffusion**”, Litu Rout, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, Wen-Sheng Chu, To appear in CVPR 2024

Recall: Diffusion Posterior Sampling (DPS)



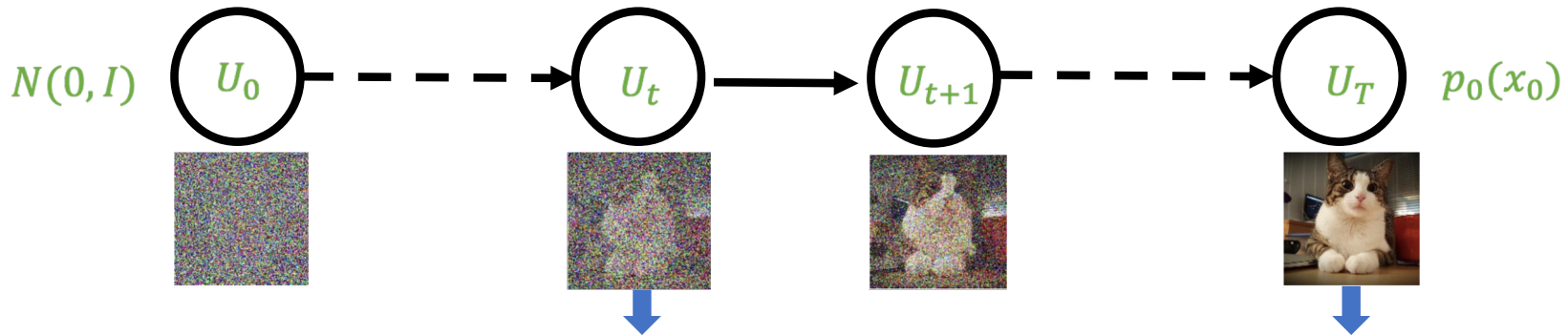
At time t we need $p_{T-t}(y|U_t)$; what we can compute instead is $p_{T-t}(y|U_T)$

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

$$p_{T-t}(y|u_t) = E_{U_T \sim p_{T-t}(U_T|u_t)}[p_0(y|U_T)] \approx p_0\left(y|E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T]\right)$$

Interpretation: Suppose $g(\cdot) = p_0(y|\cdot)$. Then, $E[g(\cdot)] \approx g(E[\cdot])$. We can compute $E[\cdot]$ using Tweedie's formula.

Interpret DPS using Taylor Series Expansion



At time t we need $p_{T-t}(y|U_t)$; what we can compute instead is $p_0(y|U_T)$

Denote $\bar{U}_T := E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T]$. Using first-order Taylor series expansion

$$\begin{aligned} p_{T-t}(y|u_t) &= E_{U_T \sim p_{T-t}(U_T|u_t)}[p_0(y|U_T)] \\ &\approx E_{U_T \sim p_{T-t}(U_T|u_t)}[p_0(y|\bar{U}_T) + \langle \nabla p_0(y|\bar{U}_T), U_T - \bar{U}_T \rangle] \\ &= p_0(y|\bar{U}_T) + \left\langle \nabla p_0(y|\bar{U}_T), E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T] - \bar{U}_T \right\rangle \end{aligned}$$

$$p_{T-t}(y|u_t) \approx p_0(y|\bar{U}_T) = p_0(y|E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T])$$

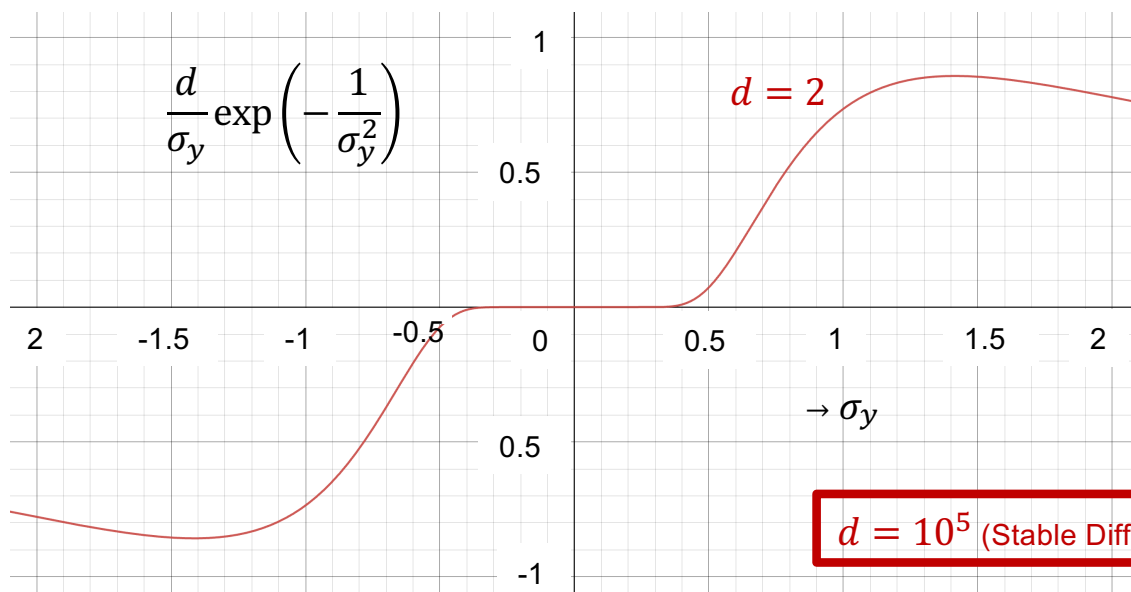
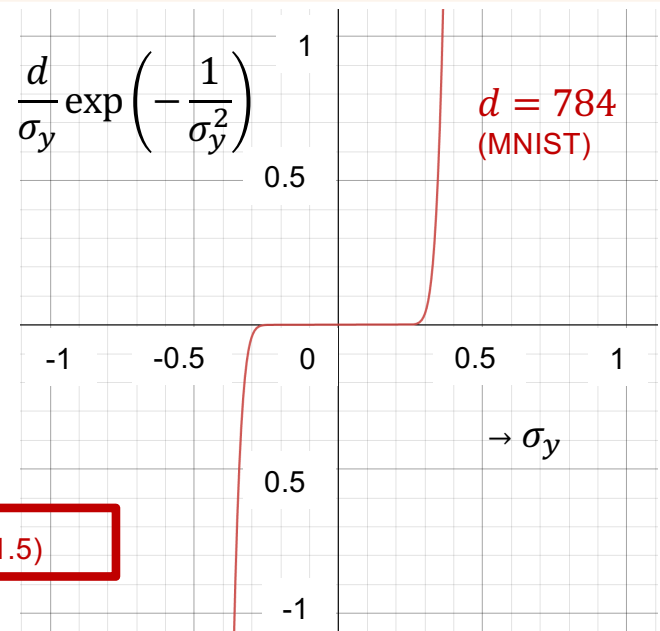
DPS Approximation Error

Theorem (Chung et al. , ICLR'23)

Given measurements $y = Ax_0 + n$ and the first-order approximation

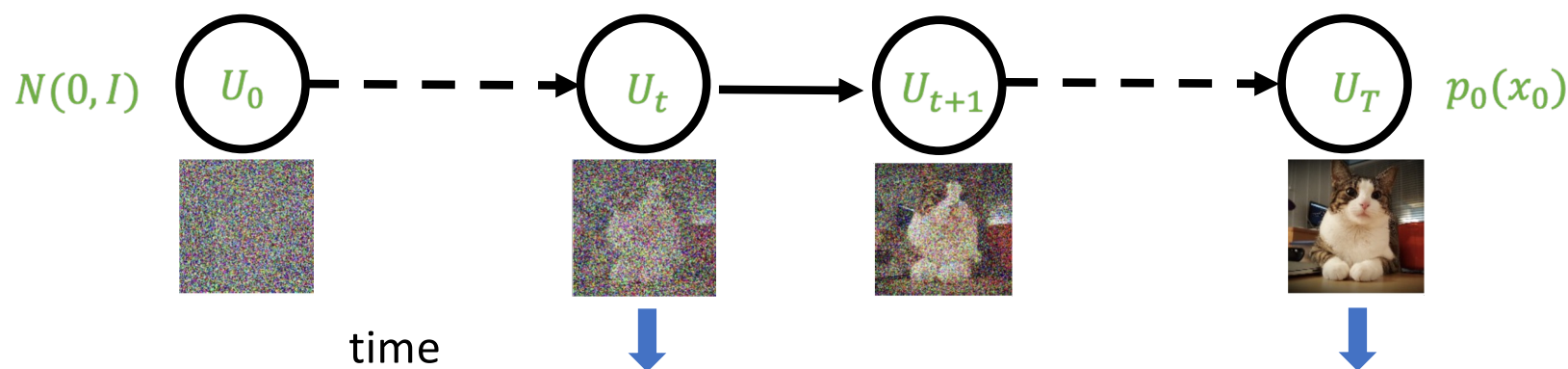
$p_{T-t}(y|u_t) \approx p_0(y|E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T])$, the error gap is upper bounded by

$$\left| E_{U_T \sim p_{T-t}(U_T|u_t)}[p_0(y|U_T)] - p_0(y|E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T]) \right| \leq O\left(\frac{d}{\sigma_y} \exp\left(-\frac{1}{\sigma_y^2}\right)\right).$$



$d = 10^5$ (Stable Diffusion v-1.5)

STSL: Second-order Tweedie from Surrogate Loss



At time t we need $p_{T-t}(y|U_t)$; what we can compute instead is $p_0(y|U_T)$

Recall Tweedie's Formula:

$$\text{Mean: } E_{U_T \sim p_{T-t}(U_T|u_t)}[U_T] = \frac{1}{\sqrt{\bar{\alpha}_{T-t}}} \left(u_t + \frac{(1-\bar{\alpha}_{T-t})}{\sqrt{\bar{\alpha}_{T-t}}} \nabla \log p_{T-t}(u_t) \right)$$

$$\text{Covariance: } E_{U_T \sim p_{T-t}(U_T|u_t)}[(U_T - \bar{U}_T)(U_T - \bar{U}_T)^T] = \frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}} (I + (1 - \bar{\alpha}_{T-t}) \nabla^2 \log p_{T-t}(u_t))$$

STSL: Second-order Tweedie from Surrogate Loss

Using second-order Taylor series expansion,

$$\log p_{T-t}(y|u_t) \geq \log p_0(y|\bar{U}_T) + \log(1 - m E_{U_T} [(U_T - \bar{U}_T)^T (U_T - \bar{U}_T)])$$

$$= \log p_0(y|\bar{U}_T) + \log\left(1 - \frac{1 - \bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}} md - (1 - \bar{\alpha}_{T-t})m \text{Trace}(\nabla^2 \log p_{T-t}(u_t))\right)$$

$$:= \hat{L}(y, u_t) \quad (\text{Surrogate loss})$$

$$\nabla \hat{L}(y, u_t) \simeq -\lambda \nabla ||y - A\bar{U}_T(u_t)||^2 - \gamma \nabla \text{Trace}(\nabla^2 \log p_{T-t}(u_t))$$

Measurement
Consistency

Second-order
Correction

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$   $\triangleright$  forward process
2 for  $t = 0$  to  $T - 1$  do
3   |  $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$   $\triangleright$  reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8     |  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  stochastic averaging
9     |  $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10    |  $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11    |  $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12    |  $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
13    |  $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}}(1 - \bar{\alpha}_{T-t-1})}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}}(1 - \alpha_{T-t})}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
14  end
15 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $\text{Enc}(\mathbf{A}^T \mathbf{y})$ and gather forward latents

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$   $\triangleright$  forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$   $\triangleright$  reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11     $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12  end
13   $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
14   $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}}(1 - \bar{\alpha}_{T-t-1})}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}}(1 - \alpha_{T-t})}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
15 end
16 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $\text{Enc}(\mathbf{A}^T \mathbf{y})$ and gather forward latents
- Initialize the reverse process at the end of the forward process

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$   $\triangleright$  forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$   $\triangleright$  reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(Z_T)\|$ 
11     $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12     $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
13     $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}}(1 - \bar{\alpha}_{T-t-1})}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}}(1 - \alpha_{T-t})}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
14  end
15 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $\text{Enc}(\mathbf{A}^T \mathbf{y})$ and gather forward latents
- Initialize the reverse process at the end of the forward process
- Use [first-order Tweedie's formula](#) to compute $\bar{Z}_T = E[Z_T | Z_t]$

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$   $\triangleright$  forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$   $\triangleright$  reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11     $-(\eta/d) \nabla (\epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t))$ 
12  end
13   $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
14   $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}}(1 - \bar{\alpha}_{T-t-1})}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}}(1 - \alpha_{T-t})}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
15 end
16 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $\text{Enc}(\mathbf{A}^T \mathbf{y})$ and gather forward latents
- Initialize the reverse process at the end of the forward process
- Use first-order Tweedie's formula to compute $\bar{Z}_T = E[Z_T | Z_t]$
- Evolve reverse process through gradient step incorporating measurement consistency

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$  ▷ forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$  ▷ reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  ▷ stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11     $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12  end
13   $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
14   $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}}(1 - \bar{\alpha}_{T-t-1})}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}}(1 - \alpha_{T-t})}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
15 end
16 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $\text{Enc}(\mathbf{A}^T \mathbf{y})$ and gather forward latents
- Initialize the reverse process at the end of the forward process
- Use first-order Tweedie's formula to compute $\bar{Z}_T = E[Z_T | Z_t] + \text{measurement consistency}$
- Gradient step including measurement consistency + **single sample update to estimate the trace of the Hessian** (Hutchinson's trace estimator + random projection to estimate higher derivatives)

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$   $\triangleright$  forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$   $\triangleright$  reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11     $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12  end
13   $Z_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
14   $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}}(1 - \bar{\alpha}_{T-t-1})}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}}(1 - \alpha_{T-t})}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
15 end
16 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $Enc(A^T y)$ and gather forward latents
- Initialize the reverse process at the end of the forward process
- Use first-order Tweedie's formula to compute $\bar{Z}_T = E[Z_T | Z_t] + \text{measurement consistency}$
- Gradient step including measurement consistency + single sample trace estimator of the Hessian + **pathwise stochastic averaging with small step size (two time-scales, with outer reverse process and inner stochastic averaging)**

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$   $\triangleright$  forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$   $\triangleright$  reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11       $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12     $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
13     $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{t+1}}(1 - \bar{\alpha}_{T-t})}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{1 - \alpha_{t+1}}(1 - \bar{\alpha}_{T-t})}{1 - \bar{\alpha}_{T-t}} Z_T$ 
14  end
15 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $\text{Enc}(\mathbf{A}^T \mathbf{y})$ and gather forward latents
- Initialize the reverse process at the end of the forward process
- Use first-order Tweedie's formula to compute $\bar{Z}_T = E[Z_T | Z_t] + \text{measurement consistency}$
- Gradient step including measurement consistency + single sample trace estimator of the Hessian + pathwise stochastic averaging
- Refine Tweedie estimate using [second-order Tweedie from surrogate loss](#)

STSL: Implementation

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$   $\triangleright$  forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$   $\triangleright$  reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11     $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12  end
13   $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
14   $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}(1 - \bar{\alpha}_{T-t-1})}}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}(1 - \alpha_{T-t})}}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
15 end
16 return  $\mathcal{D}(Z_T)$ 

```

Parsing STSL for Image Inversion

- Initialize the forward process with $\text{Enc}(A^T y)$ and gather forward latents
- Initialize the reverse process at the end of the forward process
- Use first-order Tweedie's formula to compute $\bar{Z}_T = E[Z_T | Z_t] + \text{measurement consistency}$
- Gradient step including measurement consistency + single sample trace estimator of the Hessian + pathwise stochastic averaging
- Refine Tweedie estimate using second-order Tweedie from surrogate loss
- Update latent variable in the outer reverse process

STSL: Second-order Tweedie from Surrogate Loss

- Practically Implementable
- Reduce first-order approximation error
- Requires lighter computations than
 - Jacobian of first-order score
 - Re-training of second-order score
- More efficient compared to prior works
 - 10x faster than P2L [1]
 - 3x faster than PS�D

Algorithm 1: STSL for Image Inversion

Input: diffusion time steps T , observed $\mathbf{y}$, measurement operator $\mathbf{A}$, likelihood strength λ , stochastic averaging steps K , second-order correction stepsize η , encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\mathbf{s}_\theta$

```

1 Initialization:  $\vec{Z}_0 = \mathcal{E}(\mathbf{A}^T \mathbf{y})$  ▷ forward process
2 for  $t = 0$  to  $T - 1$  do
3    $\vec{Z}_{t+1} \leftarrow \sqrt{\alpha_t} \vec{Z}_t - \sqrt{(1 - \alpha_t)} \sqrt{(1 - \bar{\alpha}_t)} \mathbf{s}_\theta(\vec{Z}_t, t)$ 
4 end
5 Initialization:  $Z_0 = \vec{Z}_T$  ▷ reverse process
6 for  $t = 0$  to  $T - 1$  do
7   for  $k = 0$  to  $K$  do
8      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  ▷ stochastic averaging
9      $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
10     $Z_t \leftarrow Z_t - \lambda \nabla \|\mathbf{y} - \mathbf{A} \mathcal{D}(\bar{Z}_T)\|$ 
11     $-(\eta/d) \nabla \left( \epsilon^T \mathbf{s}_\theta(Z_t + \epsilon, T - t) - \epsilon^T \mathbf{s}_\theta(Z_t, T - t) \right)$ 
12  end
13   $\bar{Z}_T \leftarrow (Z_t + (1 - \bar{\alpha}_{T-t}) \mathbf{s}_\theta(Z_t, T - t)) / \sqrt{\bar{\alpha}_{T-t}}$ 
14   $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}(1 - \bar{\alpha}_{T-t-1})}}{1 - \bar{\alpha}_{T-t}} Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}(1 - \alpha_{T-t})}}{1 - \bar{\alpha}_{T-t}} \bar{Z}_T$ 
15 end
16 return  $\mathcal{D}(Z_T)$ 

```

STSL: Second-order Tweedie from Surrogate Loss

- Boys *et al.*'23 and Meng *et al.*'21 show benefits of higher order moments
- Impractical due to prohibitive computational costs of $d \times d$ Hessian
 - $O(d^2)$ to compute Jacobian of the first-order score (Boys *et al.*'23)
 - Even more expensive re-training of second-order score (Meng *et al.*'21)

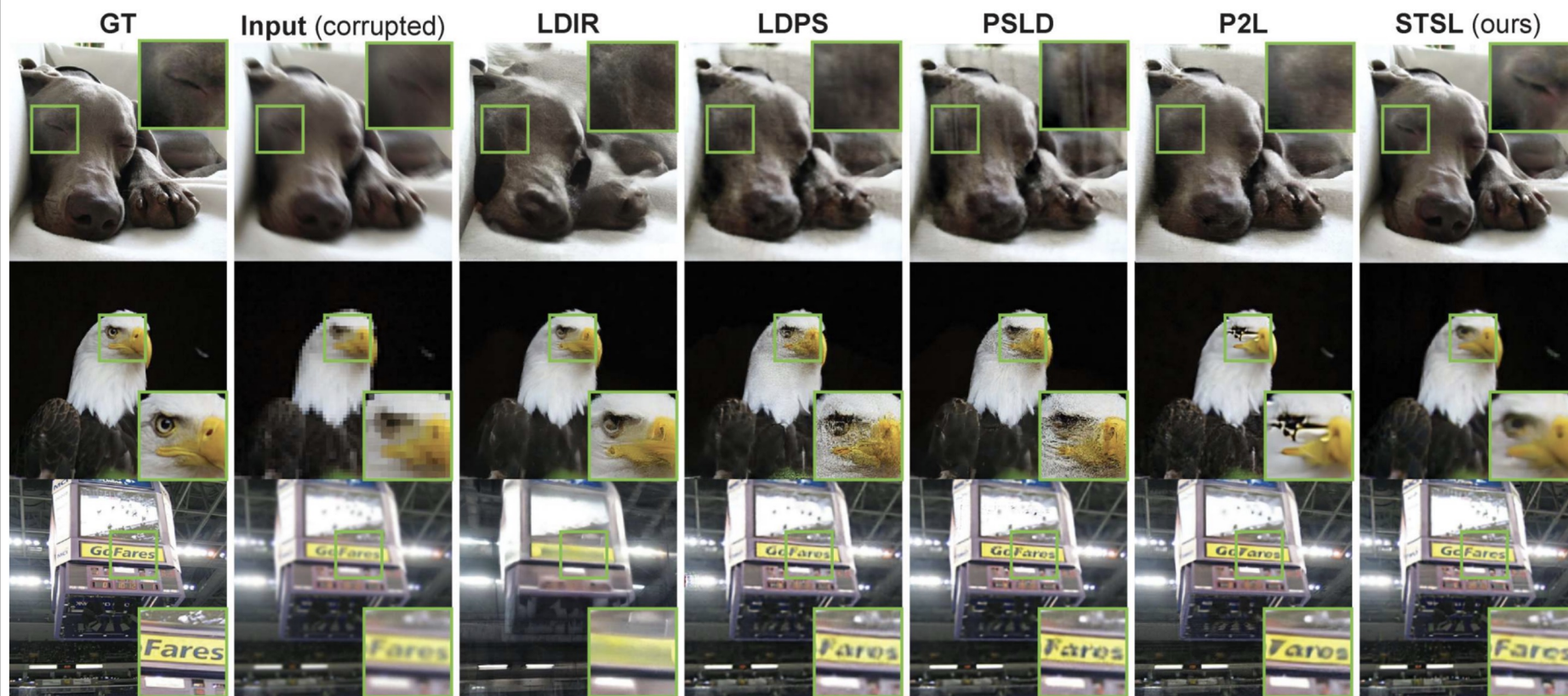
STSL requires only trace (a scalar quantity), leading to much lighter computations

STSL (Rout et al., CVPR'2024):

$$\nabla \hat{L}(y, Z_t) = \nabla \log p_{T-t}(y | \text{Dec}(\bar{Z}_T)) - \gamma \nabla \text{Trace}(\nabla^2 \log p_{T-t}(Z_t))$$

$$\text{where } \bar{Z}_T = E_{Z_T \sim p_{T-t}(Z_T | Z_t)}[Z_T]$$

Experimental Results: ImageNet



First row: Motion Deblur, **Second row:** Super-resolution, **Third row:** Gaussian Deblur.

Experimental Results: FFHQ

Method	SR ($\times 8$)			Motion Deblur			Gaussian Deblur		
	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
STSL (Ours)	0.335	31.77	91.32	0.321	31.71	90.01	0.308	<u>32.30</u>	94.04
P2L [10]	0.381	31.36	89.14	<u>0.395</u>	31.37	<u>88.81</u>	0.382	31.63	90.89
PSLD [42]	0.402	31.39	88.89	0.408	31.37	87.61	0.371	32.26	92.63
GML-DPS [42]	0.368	<u>32.34</u>	<u>92.69</u>	0.408	<u>31.43</u>	87.99	<u>0.370</u>	32.33	<u>92.65</u>
LDPS [42]	<u>0.354</u>	32.54	93.46	0.433	31.21	86.29	0.385	32.18	92.34
LDIR [15]	0.423	30.98	86.77	0.446	30.33	80.05	0.421	30.78	84.65
DPS [8]	0.538	29.15	72.92	0.556	28.98	71.95	0.694	28.14	56.15
DiffPIR [54]	0.791	28.12	42.62	0.593	28.92	67.09	0.614	29.06	73.51

- Better generative prior (e.g. Stable Diffusion) leads to better performance in posterior sampling
 - DPS and DiffPIR use pixel-space diffusion models
 - Other methods use latent diffusion model, Stable Diffusion V1.4
- Proposed algorithm can leverage the power of these strong priors

Experimental Results

Method	Runtime(s)	NFEs	Steps
STSL (ours)	45	250	50
P2L [10]	500	2000	1000
PSLD [42]	194	1000	1000
LDPS [42]	190	1000	1000
DPS [8]	180	1000	1000
DMPS [32]	67	1000	1000

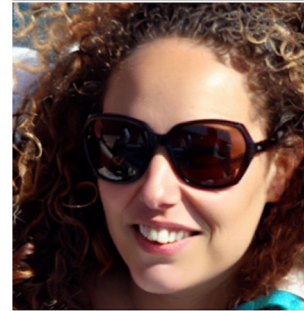
- Runtime comparable to efficient first-order methods in pixel space
- Reduction in Neural Function Evaluations (NFEs) compared to P2L (8X) and PS�D (4X)
- Outperforms state-of-the-art LDM-based solvers: P2L and PS�D

General Inverse Problems with Text Guidance

General Inverse Problems

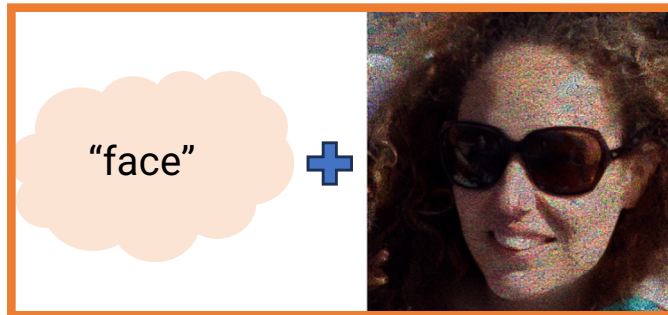


Input



Output

Text-guided Inverse Problems



Input

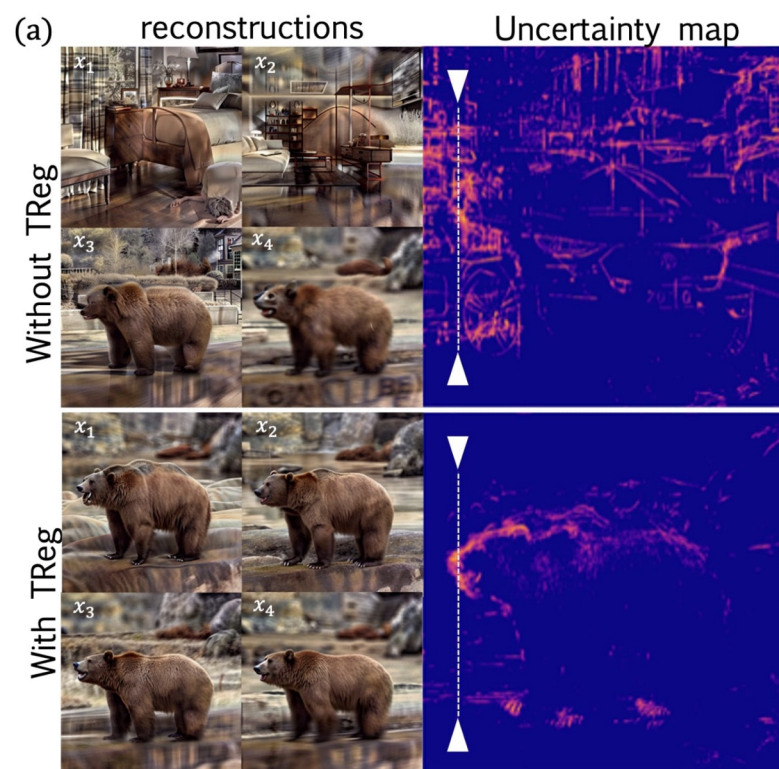
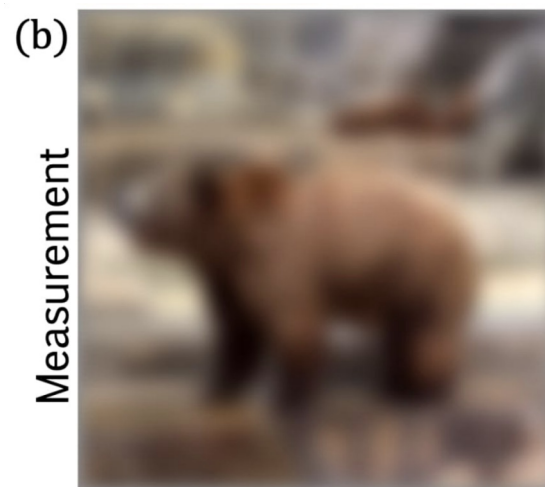


Output

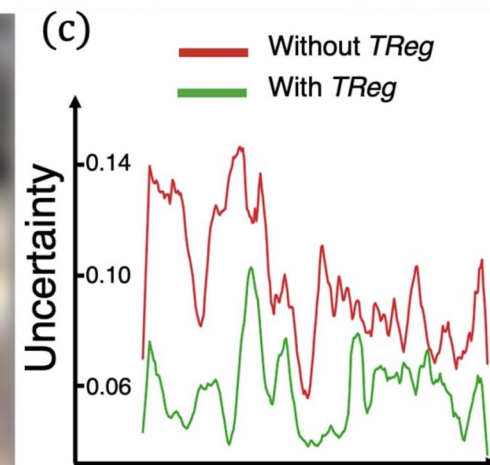
"face with red hair"



General Inverse Problems with Text Guidance

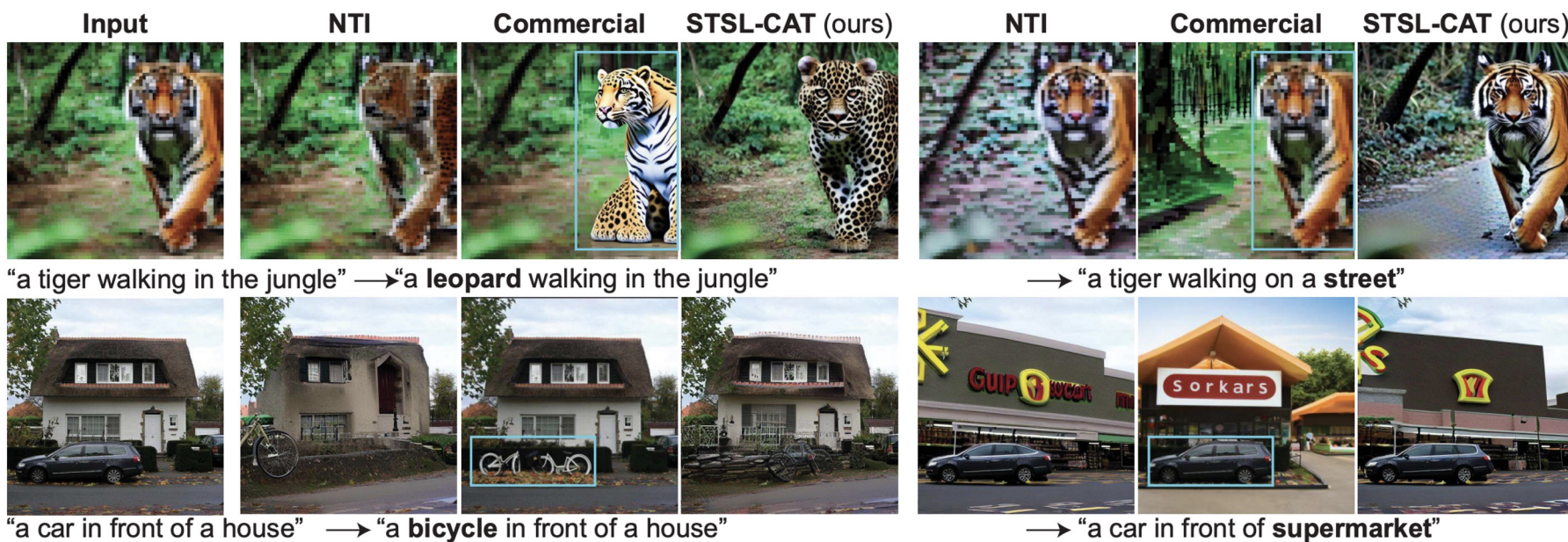


Text-prompt: "bear"



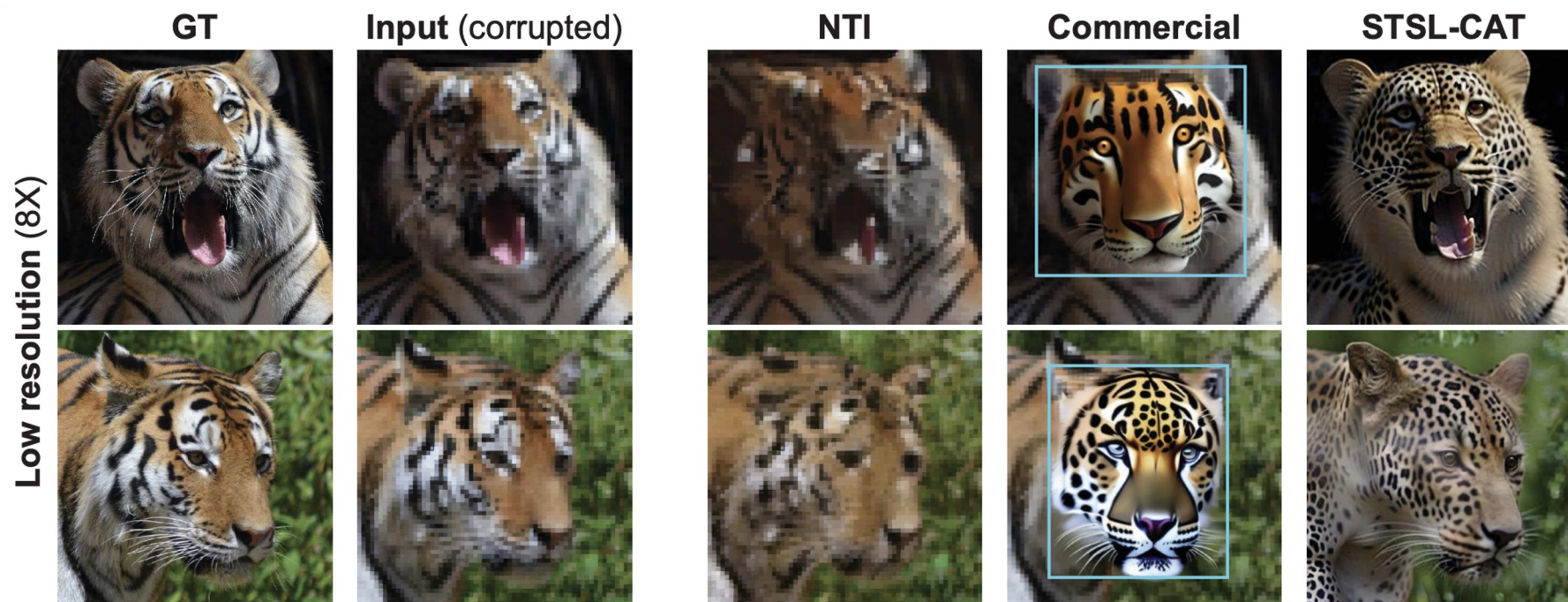
Text regularization (TReg) reduces ambiguity in the restored image

STSL for Inversion with Text Guidance



STSL can invert both corrupted (top) and clean images (bottom) with a given prompt

STSL for Inversion of Noisy Images with Text Guidance

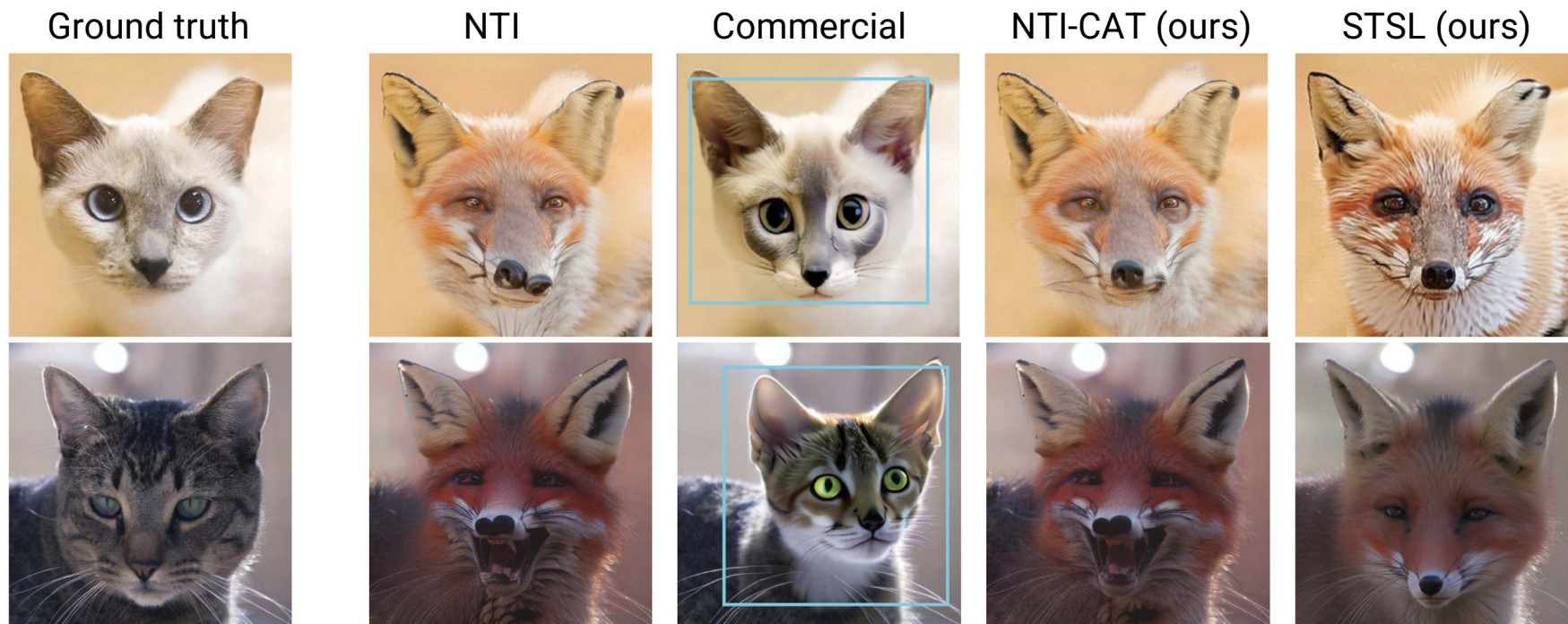


“a high quality photo of a tiger face” → “a high quality photo of a **leopard** face”

STSL for image inversion with **inconsistent** text guidance

Related work TReg by Kim et al. requires a text description that is consistent with the measurements, which might be hard to find in practice. In contrast, STSL works with consistent and inconsistent text-guidance.

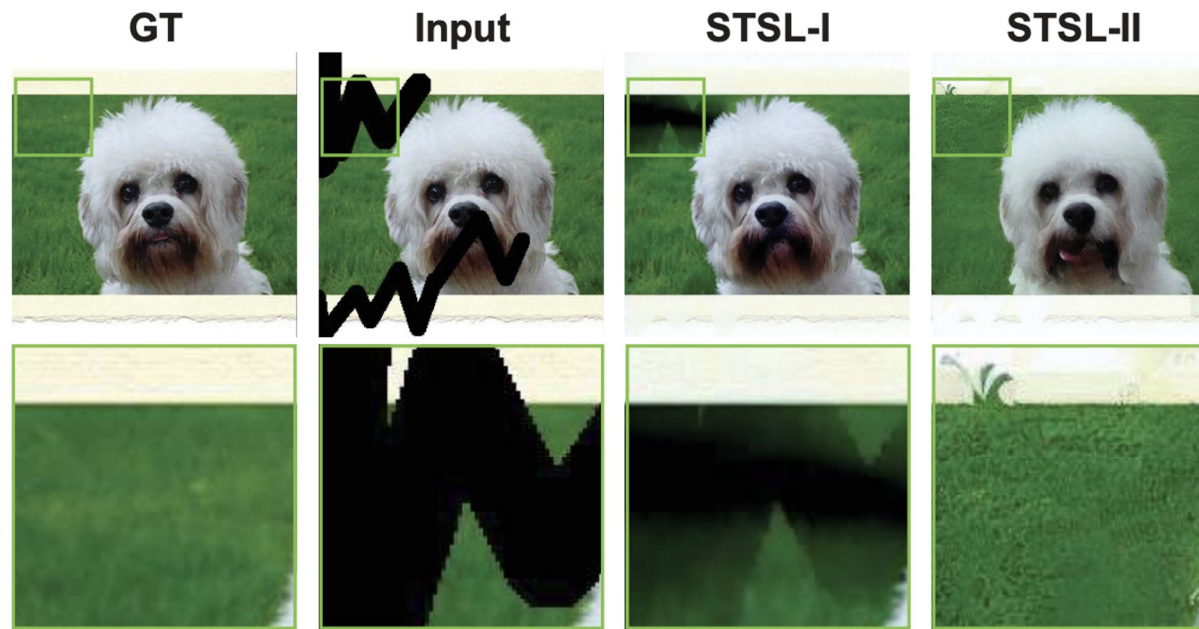
STSL for Inversion of Clean Images with Text Guidance



“a high quality photo of a cat face” → “a high quality photo of a **fox** face”

STSL preserves contents while simultaneously applying the desired edits

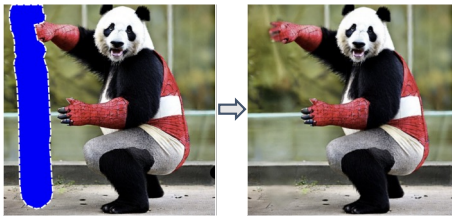
Limitations



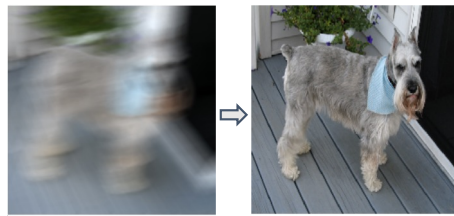
- STSL-I: Residual distortion when a large block of pixels is missing
- STSL-II: Random initialization and increasing the number of diffusion steps reduces the distortion, but becomes extremely slow

Summary

General Inverse Problems



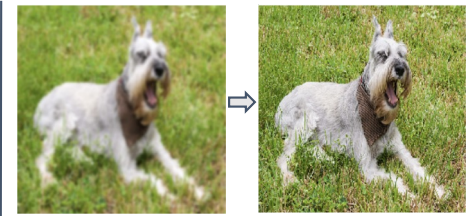
Free-form inpainting



Motion Deblur

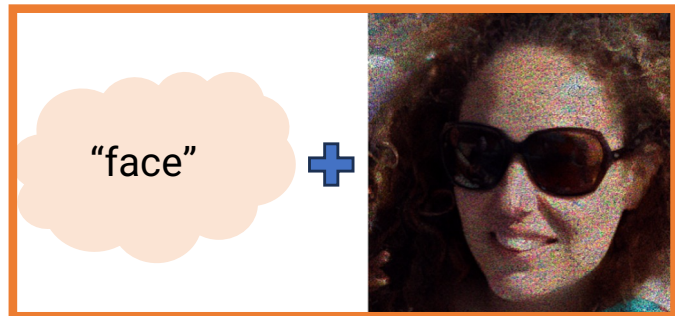


Super-resolution (4X,8X)



Gaussian Deblur

Text-guided Inverse Problems



Input



Output



"face with red hair"

Papers

“Solving Linear Inverse Problems Provably using Posterior Sampling with Latent Diffusion Models”

Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G. Dimakis, and Sanjay Shakkottai

NeurIPS 2023

“Beyond First-Order Tweedie: Solving Inverse Problems using Latent Diffusion”

Litu Rout, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, Wen-Sheng Chu

To appear in CVPR 2024