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

Litu Rout[1,2,*] Yujia Chen[1] Abhishek Kumar[3]
Constantine Caramanis[2] Sanjay Shakkottai[2] Wen-Sheng Chu[1]
[1]Google, [2]UT Austin, [3]Google DeepMind
{litu.rout,constantine,sanjay.shakkottai}@utexas.edu {yujiachen,abhishk,wschu}@google.com

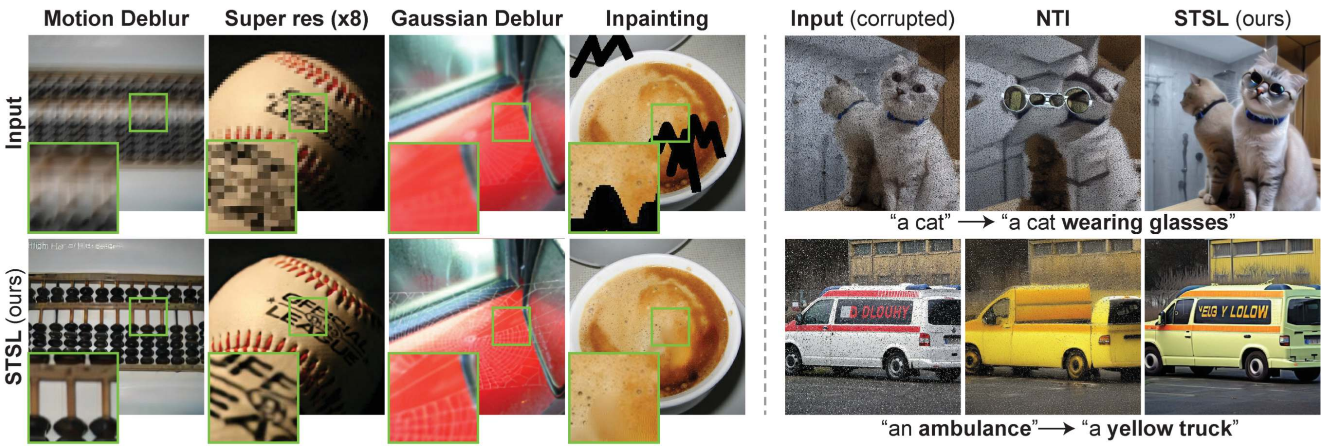

Figure 1. **Image inversion and editing with latent diffusion:** Our method, termed STSL, provides efficient inversion while enhancing the quality of reconstructed images, especially when addressing corruptions (*e.g.*, blurriness, low resolution, noise). We show its versatility in various inversion tasks (**left figure**): motion deblurring, super-resolution, gaussian deblurring, and inpainting. In addition, STSL extends to text-guided image editing with corrupted images (**right**), surpassing the performance of NTI [35], a prominent method in this domain.

Abstract

Sampling from the posterior distribution in latent diffusion models for inverse problems is computationally challenging. Existing methods often rely on Tweedie's first-order moments that tend to induce biased results [32]. Second-order approximations are computationally prohibitive, making standard reverse diffusion processes intractable for posterior sampling. We present Second-order Tweedie sampler from Surrogate Loss (STSL), a novel sampler offering efficiency comparable to first-order Tweedie while enabling tractable reverse processes using second-order approximation. Theoretical results reveal that our approach establishes a lower bound through a surrogate loss and enables a tractable reverse process using the trace of the Hessian with only $\mathcal{O}(1)$ compute. We show STSL outperforms SoTA solvers PSLD [43] and P2L [10] by reducing neural function evaluations by 4X and 8X, respectively, while enhancing sampling quality on FFHQ, ImageNet, and COCO benchmarks. Moreover, STSL extends to text-guided image editing, effectively mitigating residual distortions in corrupted images. To our best knowledge, this is the first work to offer an efficient second-order approximation for solving inverse problems using latent diffusion, which further enables editing real-world images with corruptions.

1. Introduction

This paper focuses on solving inverse problems using pre-trained latent diffusion models. The goal of linear inverse problem solvers is to find an image $\mathbf{x} \in \mathbb{R}^d$ that satisfies $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, $\mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\mathbf{y}}^2 \mathbf{I}_d\right)$[1], where $\mathbf{A} \in \mathbb{R}^{k \times d}$ is a known measurement operator and $\mathbf{y} \in \mathbb{R}^k$ is a noisy observation with unknown σ_y^2. This gives rise to a sampling challenge, where the objective is to sample the posterior $p(X|Y = \mathbf{y})$. Diffusion models are gaining popularity as priors ($p_t(X_t)$) for solving inverse problems [7, 8, 10, 43, 49]. However, the likelihood term ($p_t(\mathbf{y}|X_t)$) is only available for time $t = 0$, but not for $t > 0$, making posterior sampling inconsistent with the Bayesian posterior. One way to address this issue is training a noise conditional likelihood model, yet this is limited by training costs and the need for re-training when the measurement operator $\mathbf{A}$ changes [13]. State-of-the-art methods, such as PSLD [43] and P2L [10], resort to alternatives for computing $p_t(\mathbf{y}|X_t)$. Among these methods,

[1]**Notations.** Bold capital letter $\mathbf{X}$, bold small letter $\mathbf{x}$, normal capital letter X, and normal small letter x denote a matrix, a vector, a vector-valued random variable, and a scalar respectively. The dimension of the identity matrix $\mathbf{I}$ should be apparent from the context.
* This work was done during an internship at Google.

Tweedie's formula with first-order moments is commonly used to obtain a conditional expectation of the clean image (X_0) given the noisy image (X_t), *i.e.*, $\mathbb{E}_{X_0\sim p_t(X_0|X_t)}[X_0]$. The expected clean image is then used to approximate the likelihood as $p_t(\mathbf{y}|X_t) = \mathbb{E}_{X_0\sim p_t(X_0|X_t)} p_t(\mathbf{y}|X_0) \approx p_t(\mathbf{y}|\mathbb{E}_{X_0\sim p_t(X_0|X_t)}[X_0])$. This step reduces inconsistency in posterior sampling using Bayesian posterior [8, 10, 43].

Samplers relying on Tweedie's first-order moments are prone to sub-optimal performance due to biases in reconstruction [22, 24, 32, 34]. Recent efforts have aimed to address this bias and improve the results by introducing second-order approximation using Tweedie's formula [5, 32]. Despite these attempts, the first-order approximation is still widely used in SoTA solvers [10, 43], as existing second-order alternatives [5, 32] are hindered by significant time or memory complexity and make conventional reverse diffusion processes intractable for posterior sampling. As a result, it remains relatively unexplored to solve inverse problems with Tweedie's second-order approximation.

In this paper, we present Second-order Tweedie sampler from Surrogate Loss (STSL), introducing a new surrogate loss function to enable a tractable reverse diffusion process via efficient second-order approximation. Our key finding lies in updating the drift of the reverse process, requiring only estimates of $Trace\left(\nabla^2 \log p_t\left(\mathbf{x}_t\right)\right)$ and can be efficiently computed through random projections of the score $\nabla \log p_t(\mathbf{x}_t)$ readily available in generative models via the denoising score matching objective [21, 53]. Using Tweedie's first- and second-order moments, we estimate the mean and covariance of the Gaussian used to approximate $p_t(X_0|X_t)$. Unlike first-order methods [8, 10, 43] that approximate $p_t(X_0|X_t) \approx \delta\left(X_0 - \bar{X}_0\right)$, where $\bar{X}_0 = \mathbb{E}_{X_0\sim p_t(X_0|X_t)}[X_0]$, our STSL sampler serves as a more effective alternative because it better approximates $p_t(X_0|X_t)$. Empirical results demonstrate that STSL solves linear inverse problems in ∼50 diffusion steps, a substantial improvement over SoTA solvers [10, 43] requiring 1000 steps. This translates into 4X and 8X improvement in terms of neural function evaluations over PSLD [43] and P2L [10], respectively. We show superior performance in denosing, inpainting, super-resolution, Gaussian deblurring, and motion deblurring tasks on standard benchmarks: FFHQ [25], ImageNet [12], and COCO [30].

Using our STSL sampler, we consider image editing as another application—sampling from the posterior $p_0(X_0|\mathbf{y})$, where $\mathbf{y}$ is an input image, to obtain an edited image $X_0 \sim p_0(X_0|\mathbf{y})$. Current methods either fine-tune generative models for specific tasks [27, 44] or use a universal foundation model for all tasks [17, 35, 37]. SoTA methods, such as NTI [35], struggle with real-world corruptions [10, 43] despite excelling with clean source images, as seen in Fig. 1. Existing solvers like PSLD [43] and P2L [10] can remove corruption, but require ∼1000 diffusion steps for sampling $p_0(X_0|\mathbf{y})$, making them less practical for editing tasks. To address this, we repurpose STSL in a two-stage design: first restore the image using our inverse problem solver in just ∼50 diffusion steps, and then guide the reverse process in text-based editing using Cross-Attention-Tuning (CAT). Our results (§5) demonstrate that our approach surpasses the SoTA NTI [35] in text-guided image editing from corrupted images.

Our contributions are summarized in three-fold:

- We present an efficient second-order approximation using Tweedie's formula to mitigate the bias incurred in the widely used first-order samplers. With this method, we devise a surrogate loss function to refine the reverse process at every diffusion step to address inverse problems.
- We introduce a new framework for high-fidelity image editing in real-world environments with corruptions. To the best of our knowledge, this is the first framework that can handle corruptions in image editing pipelines.
- We conduct extensive experiments to demonstrate superior performance in tackling inverse problems (such as denoising, inpainting, super-resolution, and deblurring) and achieving high-fidelity text-guided image editing.

2. Related Work

Inverse Problems: Diffusion-based generative models are increasingly favored as effective priors for solving inverse problems, falling into two main categories: Pixel-space Diffusion Models (PDMs) [19, 47, 51] and Latent-space Diffusion Models (LDMs) [41]. While PDM-based solvers have demonstrated impressive quality [7–9, 31, 42, 49] and robustness [2, 11, 22, 23, 54] across multiple studies, they often struggle to generalize across different domains and require a specific generative model for each dataset. To overcome these limitations, recent advances such as PSLD [43] leverage the generative power of large foundation models like Stable Diffusion, outperforming PDM-based solvers by employing a single LDM for all inverse tasks.

The core concept of PSLD [43] uses the first-order Tweedie in the latent space of Stable Diffusion, *i.e.*, $\log p_{T-t}(\mathbf{y}|Z_t) \approx \log p_{T-t}(\mathbf{y}|\mathcal{D}\left(\mathbb{E}\left[Z_T|Z_t\right]\right))$. To enhance results, the latents are refined using additional gradients from a gluing objective. Building upon this, P2L [10] updates both latents and text embeddings along with a generalized gluing objective. However, both PSLD [43] and P2L [10] require a considerable amount of diffusion steps (∼1000) for satisfactory reconstruction. In contrast, our method achieves faithful reconstruction with fewer steps (∼50), offering a more practical and efficient approach.

Image Editing: As large foundation models become increasingly accessible, the realm of high-fidelity image editing emerges as a captivating domain for research. Similar to inverse problem solvers, image editing tools can be broadly classified into either PDM-based [27, 28, 44] or LDM-based

[17, 35, 37]. The former requires additional losses, such as CLIP direction loss [27], identity loss [27], structural similarity loss [28], semantic loss [28], regularization loss [44], and face preservation loss [28] to guide the reverse process in the pixel space. On the other hand, LDM-based tools [17, 35, 37] streamline the process by eliminating unnecessary complexities associated with multiple loss functions. Instead, they leverage cross-attention-control [17] on top of a text-conditional generative foundation model. However, as illustrated in Figure 1, these methods fail to apply faithful edits when confronted with real-world corruptions. Moreover, the edits are not consistently localized in the absence of corruptions. We pinpoint the fundamental cause of this failure and introduce a framework designed to address such real-world corruptions (§5).

Second-order Correction to the Tweedie Estimator: The first-order Tweedie estimator is crucial in both inverse problem solvers [8, 10, 43] and image editing tools [27, 28]. However, it tends to bias generation towards $\mathbb{E}_{Z_T \sim P(Z_T|Z_t)}[Z_T]$ instead of generating samples $Z_T \sim p_{T-t}(Z_T|Z_t)$, resulting in less detailed reconstruction. This bias is attributed to the Jensen's gap [8]. To address this limitation, we propose an alternative Tweedie estimator that only requires first-order scores, unlike prior methods [5, 32] that need second-order scores or the Jacobian of the first-order score. By focusing on the first-order score $\nabla \log p_{T-t}(\mathbf{y}|Z_t)$ when sampling $p_0(X_0|\mathbf{y})$, our estimator serves as a replacement for the Tweedie estimator with minimal computational overhead.

3. Method

3.1. Background

Diffusion Probabilistic Models (DPMs): DPMs [19, 47, 50] consist of two stochastic processes, the forward process and the reverse process. As the forward process evolves, noise is gradually added to a clean image until it becomes indistinguishable from pure noise. This progression is characterized by the general Itô Stochastic Differential Equation (SDE): $dX_t = b(X_t, t)dt + \sigma(X_t, t)dW_t$, where $X_t \in \mathbb{R}^d$, drift $b : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$, volatility $\sigma : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}$, and $\{W_t\}_{t=0}^T$ is an n-dimensional Wiener process (or Brownian motion) [36]. A simpler form of the forward process is an Ornstein-Uhlenbeck process:

$$dX_t = -X_t dt + \sqrt{2} dW_t, \quad (1)$$

which has a solution of $X_t = X_0 e^{-t} + \sqrt{2}\int_0^t e^{-(t-s)} dW_s$ that induces a Gaussian transition kernel as given by $p_t(X_t|X_0 = \mathbf{x}_0) = \mathcal{N}\left(X_t; \mathbf{x}_0 e^{-t}, (1 - e^{-2t})I\right)$. In a discrete setting, this can be written as $p_t(X_t|X_0 = \mathbf{x}_0) = \mathcal{N}(X_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)I)$, where $\bar{\alpha}_t = \prod_{s=0}^t \alpha_s$ for a finite sequence of $\alpha_s \in [0, 1]\ \forall s \in [0, T]$.

On the other hand, the reverse process gradually removes noises to produce a clean sample at the end, as characterized by the reverse Itô SDE: $dZ_t = (Z_t + 2\nabla \log p_{T-t}(Z_t))\, dt + \sqrt{2} d\tilde{W}_t$, subject to weak regularity conditions [1]. To achieve this, a neural network is trained to approximate the score $\nabla \log p_{T-t}(Z_t) \approx s_\theta(Z_t, T-t)\, \forall t \in [0, T]$ [21, 53], and the reverse Itô SDE: $dZ_t = (Z_t + 2s_\theta(Z_t, T-t))\, dt + \sqrt{2} d\tilde{W}_t$ is employed to sample from the data distribution $p_{data}(X) \coloneqq p_0(X_0)$.

Posterior Sampling: In this regime, the objective is to sample from $p_0(X_0|\mathbf{y})$ that leads to a conditional Itô SDE: $dZ_t = (Z_t + 2\nabla \log p_{T-t}(Z_t|\mathbf{y}))\, dt + \sqrt{2} d\tilde{W}_t$. Using Bayes' theorem, the drift term breaks down into $(Z_t + 2\nabla \log p_{T-t}(\mathbf{y}|Z_t) + 2\nabla \log p_{T-t}(Z_t))$. Generative foundation models, such as Stable Diffusion [41], Imagen [46], and DALL-E [38, 39] offer a reliable approximation of the true score, *i.e.*, $s_\theta(Z_t, T-t) \approx \nabla \log p_{T-t}(Z_t)$ [21, 53]. As a result, recent focus has shifted towards approximating $\log p_{T-t}(\mathbf{y}|Z_t)$. In the context of solving inverse problems or editing natural images with specific prompts, an interesting line of research [8, 10, 27, 28, 43] approximates

$$\log p_{T-t}(\mathbf{y}|Z_t) \approx \log p_{T-t}(\mathbf{y}|\mathbb{E}_{p_{T-t}(Z_T|Z_t)}[Z_T]). \quad (2)$$

We refer to (2) as the first-order Tweedie estimator for Pixel-space Diffusion Models (PDMs). For Latent-space Diffusion models (LDMs), PSLD [43] proposes the following first-order approximation (labeled as LDPS in §5):

$$\log p_{T-t}(\mathbf{y}|Z_t) \approx \log p_{T-t}(\mathbf{y}|\mathcal{D}(\mathbb{E}_{p_{T-t}(Z_T|Z_t)}[Z_T])),$$

where $\mathcal{D}(.)$ denotes a decoder from latent to pixel space. We denote the pixel-to-latent encoder as $\mathcal{E}(\cdot)$. However, methods using first-order Tweedie [8, 10, 43] introduce a bias affecting reconstruction quality [32].

Algorithm 1: STSL for Image Inversion

Input: Diffusion time steps T, observation $\mathbf{y}$, measurement operator $\mathbf{A}$, encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\boldsymbol{s}_\theta$
Tunable parameters: likelihood strength λ, stochastic averaging steps K, second-order correction stepsize η
Output: Reconstructed image $\mathcal{D}(Z_T)$

```
Compute {→Z_t}_{t=0}^T, →Z_0 ← E(A^T y)        ▷ DDIM forward proc. [35, 48]
Initialize Z_0 ← →Z_T                          ▷ proposed reverse process
for t = 0 to T − 1 do
    for k = 0 to K do
        ε ~ N(0, I)                            ▷ stochastic averaging
        Z̄_T ← (Z_t + (1 − ᾱ_{T−t}) s_θ(Z_t, T − t)) / √ᾱ_{T−t}
        Z_t ← Z_t − λ∇L(y, Z_t(Z̄_T))            ▷ Eq. (4)
    end
    Z_{t+1} ← DDIM(Z_t)                        ▷ DDIM reverse process [35, 48]
end
return D(Z_T)
```

3.2. STSL for Image Inversion

To mitigate bias in first-order Tweedie, we present Second-order Tweedie sampler from Surrogate Loss (STSL) Algorithm 1, which differs from prior methods [10, 17, 35, 43] in initialization and latent refinement. We discuss each in turn and defer implementation details to §B.1.

Initialization: Existing solvers [10, 43] initiate the reverse process from $Z_0 \sim \pi_d$, a standard Gaussian $\mathcal{N}(0, I)$, and incur a discretization error of $\mathcal{O}\left(de^{-2T}\right)$ that comes from $D_{KL}\left(p_T || \pi_d\right)$ [3, 6]. As we aim to sample $p_0(X_0|\mathbf{y})$ with *fewer diffusion steps*, this error can be substantial in *high-dimensional* sampling. To address this, we reduce the error by initializing the reverse process at $Z_0 \sim p_T(Z_0|\mathcal{E}(\mathbf{A}^T\mathbf{y}))$ and running the forward process using DDIM [35, 48] sampling, starting from $\overrightarrow{Z_0} = \mathcal{E}(\mathbf{A}^T\mathbf{y})$ (§3.1)[2]. The final latent $\overrightarrow{Z_T} \overset{\pi_d}{=} Z_0$ (equal in distribution) is then employed for initialization.

Refinement: With $Z_0 \sim p_T(Z_0|\mathcal{E}(\mathbf{A}^T\mathbf{y}))$, we propose to sample from a new reverse Itô SDE as follows:

$$dZ_t = \tilde{b}(Z_t, t)dt + \sqrt{2}d\tilde{W}_t, \tag{3}$$

where $\tilde{b}(Z_t, t) := (Z_t + \mathcal{G}(\mathbf{y}, Z_t) + 2\nabla \log p_{T-t}(Z_t))$ is the new drift. In prior works [8, 10, 43], $\mathcal{G}(\mathbf{y}, Z_t)$ represents a single gradient of the log likelihood evaluated at $\bar{Z}_T$, *i.e.* $\mathcal{G}(\mathbf{y}, Z_t) \approx \nabla \log p_{T-t}(\mathbf{y}|\bar{Z}_T) = -\lambda\nabla\|\mathbf{y} - \mathbf{A}\mathcal{D}(\bar{Z}_T)\|_2^2$, where $\bar{Z}_T := \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[Z_T\right] = \frac{Z_t}{\sqrt{\bar{\alpha}_t}} + \frac{(1-\bar{\alpha}_t)\nabla \log p_{T-t}(Z_t)}{\sqrt{\bar{\alpha}_t}}$. This correction step, essential for SoTA solvers [10, 43] after each denoising update, introduces a quality-limiting bias due to the regression to the mean $\bar{Z}_T$, as illustrated in §5.1.

In contrast, we propose an update rule that considers a second order correction term using a surrogate loss function alongside a proximal gradient update. Specifically, we approximate $\mathcal{G}(\mathbf{y}, Z_t)$ as $-\nabla\mathcal{L}(\mathbf{y}, Z_t)$, with $\mathcal{L}(\cdot, \cdot)$ denoting the surrogate loss function:

$$\mathcal{L}(\mathbf{y}, Z_t) := \lambda\|\mathbf{y} - \mathbf{A}\mathcal{D}(\bar{Z}_T)\|_2^2 \tag{4}$$
$$+ \frac{\eta}{d}\mathbb{E}_{\epsilon\sim\pi_d}\left[\epsilon^T\left(\nabla \log p_{T-t}\left(Z_t + \epsilon\right) - \nabla \log p_{T-t}\left(Z_t\right)\right)\right],$$

where the former term captures measurement consistency and the latter efficiently approximates the Hessian to mitigate biases in first-order Tweedie. To ensure we recover X_0 at the end of this new reverse process in Eq. (3), we optimize for Z_t within a small neighborhood around the corresponding forward latent $\overrightarrow{Z}_{T-t}$, which was sampled during the *Initialization* step outlined above. This is equivalent to an iterative proximal gradient update using $\mathcal{L}(\mathbf{y}, Z_t) + \kappa\|Z_t - \overrightarrow{Z}_{T-t}\|_2^2$ for some $\kappa > 0$.

Implementation: Instead of solving a local optimization, we opt for a simpler strategy by iterative gradient updates using $\mathcal{L}(\mathbf{y}, Z_t)$, and replacing the expectation in Eq. (4) with a single-sample estimate at each step [4, 29], *i.e.*, draw a random $\epsilon\sim\pi_d$ to perform the update. This iterative process of local updates accumulates into an estimate of the expectation due to path-wise stochastic averaging. Further, we add contrastive loss to the surrogate loss as $\mathcal{L}(\mathbf{y}, Z_t) + (\nu/d)\mathcal{L}_{ViT}\left(\mathbf{y}, \mathbf{A}\mathcal{D}(\bar{Z}_T)\right)$ [28] to improve the perceptual quality (§5.2). For simplicity, we do not include this term in Algorithm 1.

Practicality: This new reverse process is now tractable, and exhibits performance improvements over SoTA solvers. Ultimately, the solution to inversion materializes as $\mathcal{D}(Z_T)$, obtained at the end of Eq. (3). STSL (Algorithm 1) has similar complexity to PSLD (Algo. 2 in [42]), with one extra correction term η, and outperforms P2L [10] by avoiding text-embedding optimization. Despite the improvement from an extra hyper-parameter ν in the contrastive loss, STSL excels without it (Table 3). In practice, we use the same configuration across *all* the experiments on three datasets and achieve promising results, which proves robustness of STSL to hyper-parameters.

3.3. STSL for Image Editing

Inverting Corrupted Latents: To edit a real image, NTI [35] stands out as a leading method that associates the real image with a sequence of null embeddings. Formally, define $\Phi(\cdot)$ as an encoder that maps a text prompt to an embedding in $\mathbb{R}^h$. Given a text-conditional score network $s_\theta : \mathbb{R}^d \times \mathbb{R}_+ \times \mathbb{R}_h \to \mathbb{R}^d$, NTI [35] tackles the optimization problem $\hat{\varphi}_t = \arg\min_{\varphi_t}\|\overrightarrow{Z}_{T-t-1} - f\left(Z_t, T-t, \varphi_t\right)\|_2^2$, with $\{\varphi_t\}_{t=0}^T$ initialized by null-text embeddings and $f\left(Z_t, T-t, \varphi_t\right) = \sqrt{\bar{\alpha}_{T-t-1}}\bar{Z}_T + \sqrt{1-\bar{\alpha}_{T-t-1}}\sqrt{1-\bar{\alpha}_{T-t}}s_\theta(Z_t, T-t, \varphi_{T-t})$. Instead, we propose to solve: $\hat{\varphi}_t = \arg\min_{\varphi_t}\|Z_{t+1} - f\left(Z_t, T-t, \varphi_t\right)\|_2^2$, where $\{Z_t\}_{t=0}^T$ are obtained from our novel reverse SDE using the surrogate loss in Eq. (4). NTI [35] associates $\{\varphi_t\}_{t=0}^T$ with the corrupted image, leaving corruptions in the edited image (see §5.3). In contrast, our proposed null-optimization aligns $\{\varphi_t\}_{t=0}^T$ with a clean image because our new SDE in Eq. (3) yields a clean image at the end. Thereby, it enables text-guided noisy image editing via Cross-Attention-Control (CAC) [17].

Cross-Attention-Tuning (CAT): Conventional CAC-based image editing [17] encounters a critical limitation – it struggles to maintain the original image content while incorporating the desired modifications (§5.3). To address this issue, our method refines the latents after CAC update via posterior sampling. Let $\mathcal{C}$ denote the CAC module that gives $\hat{Z}_{t+1} = \mathcal{C}\left(Z_t, \Phi(\text{“prompt”}), t, \hat{\varphi}_t\right)$. Then, we update Z_{t+1} using a single step of STSL:

$$Z_{t+1} \leftarrow \hat{Z}_{t+1} - \lambda\nabla\mathcal{L}\left(\mathbf{y}, \bar{Z}_T\left(\hat{Z}_{t+1}\right)\right). \tag{5}$$

Inverse problem solvers and image editing tools differ due to the absence of ground truth measurements in the latter. This limits the effectiveness of the measurement update $\nabla\|\mathbf{y} - \mathbf{A}\mathcal{D}(\bar{Z}_T)\|_2^2$ during editing. To address this,

[2] An alternative could be to initialize $\overrightarrow{Z_0}$ as $\arg\min_Z\|\mathbf{y} - \mathbf{A}\mathcal{D}(Z)\|_2^2$. However, this is more expensive. Empirically, we observe good results using one step projection, which is significantly less expensive.

we employ contrastive loss [28] between the edited image $\mathcal{D}(\bar{Z}_T)$ and the input image's features extracted via ViT's multi-head self-attention layers [52]. To ensure meaningful features, we apply measurement updates in the initial phase (30 steps) of the reverse process, and then update the latents in Eq. (5) using contrastive loss: $Z_{t+1} \leftarrow Z_{t+1} - \frac{\nu}{d}\nabla\mathcal{L}_{ViT}\left(\mathbf{y}, \mathbf{A}\mathcal{D}(\bar{Z}_T)\right)$. The cross-attention features help preserve the image's *content* while incorporating desired *semantics*. We term this process Cross-Attention-Tuning (CAT), which forms the basis of our proposed image editing method STSL-CAT.

4. Theory

Recall from §3.1 we seek for a good approximation of $\log p_{T-t}(\mathbf{y}|Z_t)$. Methods based on Tweedie's first order moment leverage the posterior mean from **Proposition 4.1** to approximate $\nabla \log p_{T-t}(\mathbf{y}|Z_t) \approx \nabla \log p_{T-t}(\mathbf{y}|\bar{Z}_T)$, where $\bar{Z}_T = \mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t)}[Z_T]$. Since $s_\theta(Z_t, T-t) \approx \nabla \log p_{T-t}(Z_t)$, this enables a practical implementation of posterior samplers [8, 10, 43, 49].

Proposition 4.1 ([14, 40])**.** *Given* $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ *and* $\epsilon \sim \mathcal{N}(0, I)$*, denote by* $\bar{X}_0 = \mathbb{E}_{X_0 \sim p_t(X_0|X_t=\mathbf{x}_t)}[X_0]$ *the posterior mean of* $p_t(X_0|X_t = \mathbf{x}_t)$*. Then, for the variance preserving SDE in* (1) *or DDPM sampling [19],* $p_t(X_0|X_t = \mathbf{x}_t)$ *has mean* $\bar{X}_0 = \frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}} + \frac{(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\nabla_{\mathbf{x}_t}\log p_t(X_t = \mathbf{x}_t)$ *and covariance* $\mathbb{E}_{X_0 \sim p_t(X_0|X_t=\mathbf{x}_t)}\left[\left(X_0 - \bar{X}_0\right)\left(X_0 - \bar{X}_0\right)^T\right] = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\left(\mathbf{I} + (1-\bar{\alpha}_t)\nabla^2_{\mathbf{x}_t}\log p_t(X_t = \mathbf{x}_t)\right)$.

The error from the first-order approximation is characterized by the Jensen's gap [8, Theorem 1]. The gap can be notable in practice due to the local linearity of first-order approximations [32]. Prior works [5, 32] mitigate this error by a second-order approximation that introduces curvature to the estimator. However, these methods [5, 32] require expensive Hessian computations. In contrast, our sampler requires only the *trace* of the Hessian, and hence enable efficient computation with minimal cost. We present our main results in **Theorem 4.4** under the assumptions stated below.

Assumption 4.2. *Define* $\bar{Z}_T := \mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t)}[Z_T]$ *for* $t \in [0, T]$*. Then,* $p_{T-t}(\mathbf{y}|\bar{Z}_T) = \mathcal{N}\left(\mathbf{y}; \mathbf{A}\bar{Z}_T, \sigma^2_{\mathbf{y}}\mathbf{I}\right)$ *and* $p_{T-t}(\mathbf{y}|\bar{Z}_T) > 0$.

Assumption 4.3. *For all* $\tilde{Z}_T \in \mathbb{R}^d$ *and* $m > 0$*,* $-mp_{T-t}(\mathbf{y}|\bar{Z}_T)\mathbf{I} \preceq \nabla^2 p_{T-t}(\mathbf{y}|Z_t)\big|_{\tilde{Z}_T}$*, where* $\bar{Z}_T := \mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t)}[Z_T]$ *for* $t \in [0, T]$.

Assumption 4.2 is a common condition in prior works [8, 10, 42, 43], indicating informative measurements from $\bar{Z}_T$. **Assumption 4.3** simplifies mathematical considerations and ensures the smallest eigenvalue of the Hessian is uniformly lower bounded by a finite quantity. Notably, the widely used Gaussian measurement model $\mathbf{y} = \mathbf{A}\bar{Z}_T + \sigma^2_{\mathbf{y}}\mathbf{n}, \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ satisfies both these assumptions.

Theorem 4.4 (Tweedie Sampler from Surrogate Loss)**.** *Suppose* ***Assumption 4.2*** *and* ***Assumption 4.3*** *hold. Let* $\hat{\mathcal{L}}(\mathbf{y}, Z_t) := \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(\xi_t - (1-\bar{\alpha}_t)m\, Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right)$*, where* $\xi_t = 1 - \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}md$*. For* $\lambda = \mathcal{O}(\frac{1}{\sigma^2_{\mathbf{y}}})$ *and* $\gamma = \mathcal{O}(\frac{\eta}{d})$*, the following holds:* $\hat{\mathcal{L}}(\mathbf{y}, Z_t) \leq \log p_{T-t}(\mathbf{y}|Z_t)$ *and the gradient of* $\hat{\mathcal{L}}(\mathbf{y}, Z_t)$ *becomes* $\nabla\hat{\mathcal{L}}(\mathbf{y}, Z_t) \simeq -\lambda\nabla\|\mathbf{y} - \mathbf{A}\bar{Z}_T\|^2_2 - \gamma\nabla\left(Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right)$.

Proof. The proof is included in Appendix A.3. □

We draw the following insights from **Theorem 4.4. Practically Implementable:** Given $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we use Hutchinson's estimator [20], along with a random projection based gradient estimator [15] to compute the trace of the Hessian as given below $Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right) \simeq \mathbb{E}\left[\epsilon^T\left(\nabla \log p_{T-t}(Z_t + \epsilon) - \nabla \log p_{T-t}(Z_t)\right)\right] - \mathcal{O}(\|\epsilon\|^3)$ (Appendix A.4 for details).

Substituting this in **Theorem 4.4** and extending the result to LDMs, our algorithm becomes tractable because it requires $\nabla \log p_{T-t}(Z_t) \approx s_\theta(Z_t, T-t)$, which is readily available in LDMs [41]:

$$\nabla\hat{\mathcal{L}}(\mathbf{y}, Z_t) \simeq -\lambda\nabla\|\mathbf{y} - \mathbf{A}\mathcal{D}(\bar{Z}_T)\|^2 - \gamma\nabla\mathbb{E}_{\epsilon\sim\pi_d}\left[\epsilon^T\left(\nabla \log p_{T-t}(Z_t + \epsilon) - \nabla \log p_{T-t}(Z_t)\right)\right]. \tag{6}$$

Further, $\nabla\hat{\mathcal{L}}(\mathbf{y}, Z_t) \simeq -\nabla\mathcal{L}(\mathbf{y}, Z_t)$, where $\mathcal{L}(\mathbf{y}, Z_t)$ is the surrogate loss function in §3.2 (Appendix A.3 for details).
Connection with First-order Tweedie: The update in Eq. (6) samples from an alternate reverse process $dZ_t = \tilde{b}(Z_t, t)dt + \sqrt{2}d\tilde{W}_t$. Thus, in Eq. (2), if we modify the first-order Tweedie's conditional drift by setting $\gamma = 0$ and applying a one-step gradient of $\mathcal{L}(\mathbf{y}, Z_t)$, it becomes a special case of our drift $\tilde{b}(Z_t, t)$. However, this setup introduces a bias that could hamper the quality, as we illustrate in §5. To mitigate this bias, we implement multiple proximal gradient updates and leverage stochastic averaging to estimate expectations using the surrogate loss $\mathcal{L}(\mathbf{y}, Z_t)$.
Computational Complexities: Using the gradient update given in Eq. (6), our sampler provides an efficient second-order approximation. It uses a correction step, denoted as $\nabla\mathbb{E}_{\epsilon\sim\pi_d}\left[\epsilon^T\left(\nabla \log p_{T-t}(Z_t + \epsilon) - \nabla \log p_{T-t}(Z_t)\right)\right]$, which we estimate using only $\mathcal{O}(1)$ number of steps. This stands in contrast to prior methods that require $\mathcal{O}(d^2)$ compute, such as calculating the Jacobian of the score $\nabla\left(\nabla \log p_{T-t}(Z_t)\right)$ [5], or even more expensive approaches like re-training of the Hessian $\nabla^2 \log p_{T-t}(Z_t)$ [32]. Further theoretical results are deferred to Appendix A.

5. Experiments

Datasets: We adhere to FFHQ [25] and ImageNet [12] benchmarks with 512×512 images. For FFHQ, we use the

Method	SR (×8)			Motion Deblur			Gaussian Deblur			SR (×8)			Motion Deblur			Gaussian Deblur		
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
STSL (Ours)	**0.335**	31.77	91.32	**0.321**	**31.71**	**90.01**	**0.308**	32.30	**94.04**	**0.392**	**30.64**	**84.86**	**0.420**	30.38	76.45	**0.349**	31.03	**90.21**
P2L [10]	0.381	31.36	89.14	0.395	31.37	88.81	0.382	31.63	90.89	0.441	30.38	84.27	0.429	**30.65**	**84.53**	0.402	30.70	87.92
PSLD [43]	0.402	31.39	88.89	0.408	31.37	87.61	0.371	32.26	92.63	0.484	30.23	80.72	0.433	30.50	82.17	0.387	**31.20**	88.65
GML-DPS [43]	0.368	32.34	92.69	0.408	31.43	87.99	0.370	**32.33**	92.65	0.482	30.35	81.01	0.432	30.55	82.65	0.387	31.19	88.60
LDPS [43]	0.354	**32.54**	**93.46**	0.433	31.21	86.29	0.385	32.18	92.34	0.480	30.25	80.87	0.462	30.36	80.03	0.402	31.14	88.19
LDIR [16]	0.423	30.98	86.77	0.446	30.33	80.05	0.421	30.78	84.65	0.498	30.04	78.22	0.531	29.56	68.32	0.501	29.81	74.56
DPS [8]	0.538	29.15	72.92	0.556	28.98	71.95	0.694	28.14	56.15	0.522	29.18	70.72	0.548	28.79	68.86	0.647	28.11	55.40
DiffPIR [55]	0.791	28.12	42.62	0.593	28.92	67.09	0.614	29.06	73.51	0.740	28.17	42.80	0.577	28.81	61.20	0.612	28.93	68.67

Table 1. **Quantitative results of inversion:** On FFHQ-1K (512×512, **left**) and ImageNet-1K (512×512, **right**), the results are obtained with a noise level $\sigma_{\mathbf{y}} = 0.01$. Best methods are emphasized in **bold** and second best underlined. PDM-based solvers are shaded in gray; LDM-based solvers are in the middle row block. Notably, our method STSL outperforms leading inverse problem solvers [10, 43].

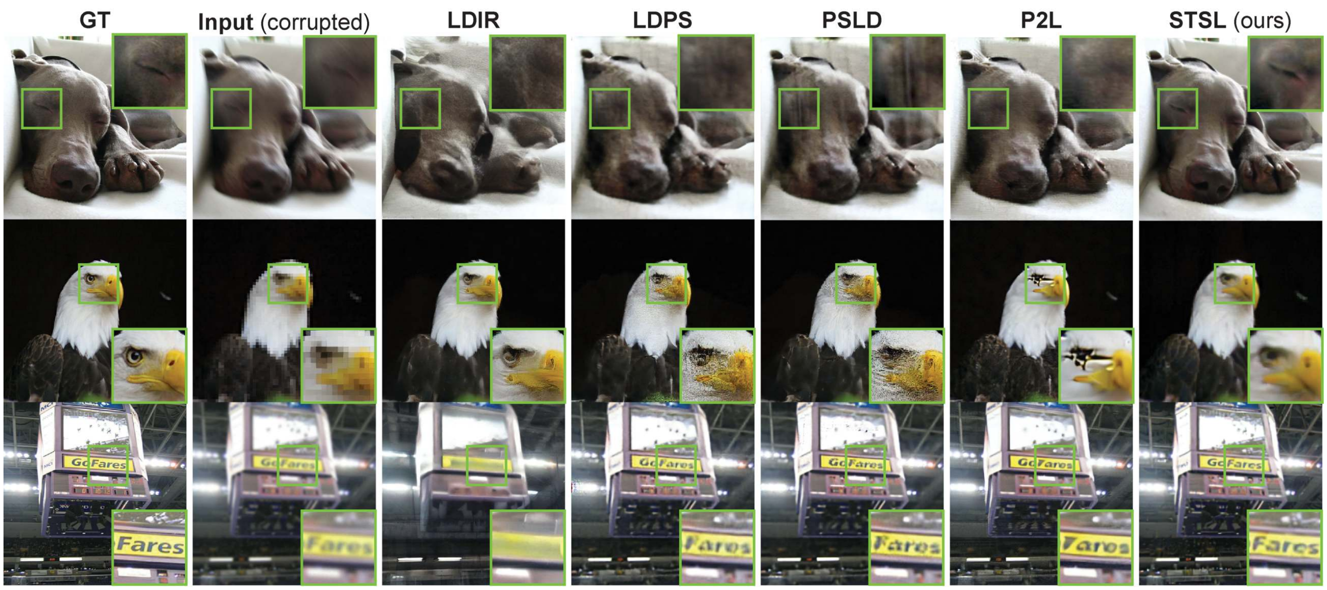

Figure 2. **Qualitative results of inversion on ImageNet (512×512):** All the compared methods utilize the same foundation model, Stable Diffusion. The top, middle, and bottom rows represent the results on Motion Deblur, Super Resolution (8X), and Gaussian Deblur, respectively. The highlighted segment distinctly reveals the superior performance of our method (**STSL**).

identical set of 1000 images as prior work [8, 10, 43]. For ImageNet, we follow P2L [10] by uniformly sampling 1000 images from the `ctest10k` split [45]. We conduct ablation studies using COCO 2017 validation set [30].

Baselines: For inverse problems, we benchmark against SoTA solvers PSLD [43] and P2L [10], alongside LDM-based methods LDPS [43], GML-DPS [43] and LDIR [16]. Note that LDPS and GML-DPS were first introduced in PSLD [43] and later extended to prompt-tuning in P2L [10]. For completeness, we also extend comparisons to PDM-based solvers DPS [8] and DiffPIR [55]. For image editing, we compare with a leading solution NTI [35].

Inverse Tasks: We investigate 5 inverse tasks: motion deblurring, super-resolution (SR), Gaussian deblurring, random inpainting, and denoising. We follow the setup of P2L [10] for motion deblurring[3], Gaussian deblurring, and super-resolution (8X). While testing SR at 8X could be ambitious, it challenges these algorithms to their limits. We also test in less demanding inverse problems, *i.e.*, SR at 4X, random inpainting with 40% dropped pixels, and denoising for salt-pepper noise with 2% noises.

Metrics: We evaluate using standard metrics: LPIPS, PSNR and SSIM. For editing, we resort to CLIP accuracy [37]. All experiments are conducted on a single A100 GPU. See Appendix B.1 for details.

5.1. Results on Image Inversion

Inversion Quality: We evaluate our method against SoTA LDM-based solvers [10, 16, 43] and PDM solvers [8, 55] on FFHQ and ImageNet datasets. Table 1 summarizes the results, highlighting STSL's ability in restoring the perceptual similarity of the original image, with improvement over SoTA P2L [10] and PSLD [43], particularly evident in LPIPS scores. Table 1 also underscores the prevailing trend in LDM-based solvers [10, 43] compared to PDM-based solvers [8, 26, 49]. Notably, STSL surpasses P2L and

[3] https://github.com/LeviBorodenko/motionblur

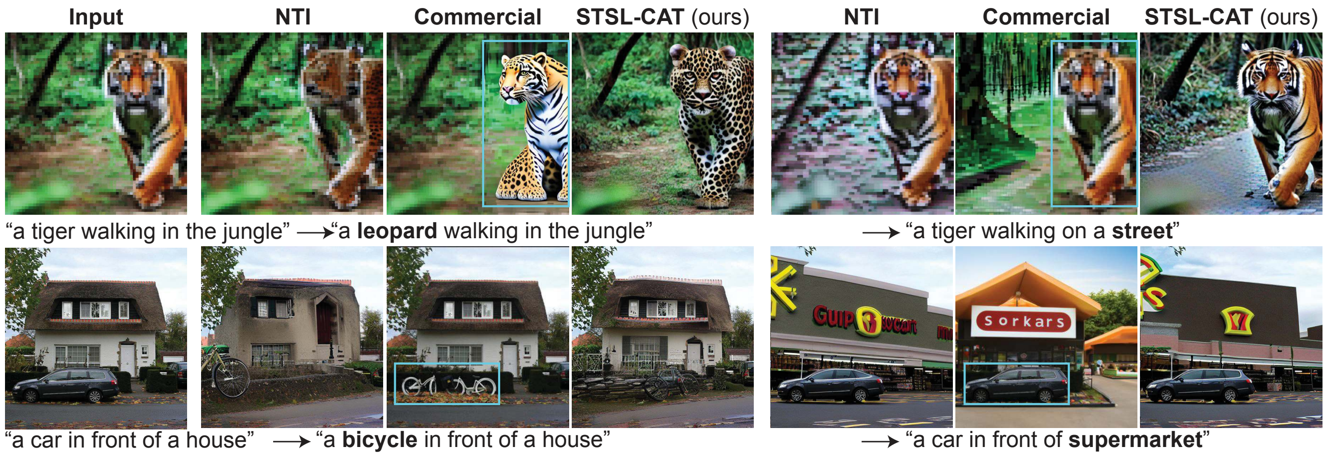

Figure 3. **Qualitative results on image editing:** We compared our method with NTI [35] and commercial software on images with corruption (SRx8 **top**) and without (**bottom**). The commercial software requires an additional mask to localize the edits (cyan rectangles).

Method	Runtime(s)	NFEs	Steps
STSL (ours)	45	250	50
P2L [10]	500	2000	1000
PSLD [43]	194	1000	1000
LDPS [43]	190	1000	1000
DPS [8]	180	1000	1000
DMPS [33]	67	1000	1000

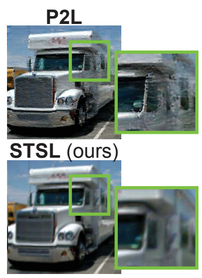

Table 2. **(left)** Efficiency of LDM (top 4 rows) and PDM solvers (bottom 2 rows) on the super-resolution 8X task. **(right)** Comparison of the image quality. The P2L image has LPIPS/SSIM of 0.51/74, and ours are 0.47/71.

PSLD with a 5% absolute improvement in LPIPS on the demanding 8x super-resolution task. Figure 2 shows that STSL produces sharper and more detailed images without introducing artifacts or hallucination.

We adopt hyperparameters following the convention of inverse problem solvers [8, 10, 43], optimizing for LPIPS as it aligns with human perception. While maintaining competitive results on PSNR/SSIM, we notice our SSIM score could be less satisfactory in some cases. Our exploration reveals SSIM's inclination to label high-frequency artifacts as "sharpness" and its tendency to penalize blurriness more. In Table 2 (right), the SSIM score of the P2L output is much better than ours regardless of the artifacts. Consequently, the following discussion primarily emphasizes LPIPS.

Inversion Efficiency: Table 2 (left) compares solver efficiency amongst SoTA methods. LDM-based solvers generate 512×512 images whereas PDMs produce 256×256. We downscale the LDM runtime by 4X for a fair comparison with PDMs. P2L [10] runtime is estimated based on its pseudo-code due to the unavailability of source code. Consequently, STSL achieves desired results in fewer diffusion steps ($T = 50$) compared to PSLD [43] and P2L [10] ($T = 1000$). This computational gain allows extra budget for local iterative gradient updates ($K = 5$) using our surrogate loss. In addition, STSL realizes a notable 4X improvement in the number of NFEs compared to PSLD [43], and 8X over P2L [10]. Since NFE is the most expensive component in posterior sampling (§3.1), less NFEs translate to practical advantages in runtime efficiency.

5.2. Ablation Study

Bias Analysis: STSL mitigates bias in the first-order Tweedie estimator by employing stochastic averaging steps with the surrogate loss $\mathcal{L}(\mathbf{y}, Z_t)$ (§3.2) and an alternative reverse process (3) initialized at $Z_0 \sim p_T(Z_0|\mathcal{E}(\mathbf{A}^T\mathbf{y}))$. By eliminating Hutchinson's trace estimator ($\eta = 0$) and using a single-step gradient update with $\mathcal{L}(\mathbf{y}, Z_t)$, we derive the biased estimator STSL-biased. The disparity between STSL-biased and PSLD [43] lies in their initialization: the former begins with Z_0 drawn from $p_T(Z_0|\mathcal{E}(\mathbf{A}^T\mathbf{y}))$, while the latter from π_d. Evaluating on 100 random images in FFHQ, Table 3 shows the advantages of our improved initialization and stochastic averaging steps over PSLD [43].

Component Analysis: We study the significance of each component in STSL using 50 random samples from the COCO [30] dataset. Hyperparameters derived from this study also generalize to FFHQ-1K [25] and ImageNet-1K [12] datasets, showing the robustness of our method in handling unseen domains. Throughout this analysis, we use salt-pepper noise as corruption in all the experiments.

Table 4(a) shows that employing a 2-sample average yields a more accurate estimate of the expectation in our surrogate loss $\mathcal{L}(\mathbf{y}, Z_t)$. With 5 stochastic averaging steps, we have $N = 10$ Gaussian samples (ϵ) in total per diffusion time step. More samples lead to longer running time and higher memory demand with marginal benefits in LPIPS, but sharper image quality as evident from PSNR/SSIM.

Table 4(b) shows the Hutchinson's trace estimator, denoted as η, plays a crucial role. Optimal results were

Method	Gaussian Deblur			Salt-pepper Denoise			SR (×4)			Random Inpaint			SR(×8)		
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
STSL (Ours)	**0.380**	**30.82**	**91.55**	**0.322**	**30.43**	**89.95**	**0.326**	**30.58**	**91.67**	**0.405**	**30.13**	**88.24**	**0.398**	**30.32**	**88.12**
STSL-biased (Ours)	0.456	30.81	90.70	0.378	30.29	88.12	0.406	30.52	89.91	0.547	29.64	83.43	0.444	30.26	85.18
PSLD [43]	0.569	29.65	75.19	0.530	29.79	78.74	0.531	29.84	80.22	0.693	28.86	67.25	0.538	29.71	77.71

Table 3. **Quantitative results for bias analysis:** All methods use 50 diffusion steps. STSL uses 5 stochastic averaging steps. STSL-biased uses a single step gradient update. The results are obtained with noise level $\sigma_{\mathbf{y}} = 0.05$.

(a)

N	LPIPS	PSNR	SSIM
1	0.408	29.45	80.09
2	**0.403**	29.46	81.83
3	0.405	**29.51**	**81.89**

(c)

ν	0	0.5	2	5	10
LPIPS	0.404	0.399	**0.392**	0.401	0.413
PSNR	**29.52**	29.44	29.39	29.31	29.19
SSIM	81.94	81.42	**82.09**	81.07	80.36

(e)

K	LPIPS	PSNR	SSIM	NFE
2	0.432	**29.51**	80.13	20
5	**0.408**	29.45	**90.09**	50
10	0.424	29.33	77.42	100

(b)

η	LPIPS	PSNR	SSIM
0	0.404	29.52	81.94
0.02	**0.388**	29.52	**82.70**
0.05	0.395	**29.54**	82.45

(d)

Steps	5	10	50
LPIPS	0.564	0.511	**0.408**
PSNR	28.56	28.87	**29.45**
SSIM	66.91	73.57	**80.09**

(f)

Method	NTI (clean)	NTI-CAT	NTI	STSL-CAT
LPIPS	0.434	**0.425**	0.638	**0.552**
PSNR	29.06	**29.11**	28.74	**28.81**
CLIP acc.	**96.00**	**96.00**	70.00	**93.00**

Table 4. **Ablation studies**: (a) the number of samples $\epsilon \sim \pi_d$ used in stochastic averaging (§3.2) at each diffusion step, (b) coefficient of the second-order approximation term in Eq. (4) with a single ϵ, (c) coefficient of the contrastive loss, (d) the number of DDIM steps, (e) the number of the stochastic averaging steps, and (f) image editing results compared with NTI [35], where the first two columns "NTI (clean)" and "NTI-CAT" are with clean images to show the effectiveness of CAT, while the last two column are on corrupted images with SRx8.

achieved at $\eta = 0.02$. A value too small fails to mitigate the bias discussed in §3.2, while too large values deviate the forward process, resulting in a different image.

Table 4(c) demonstrates the effectiveness of the contrastive loss controlled by ν introduced in §3.2. As discussed in §3.2 and §3.3, the contrastive loss helps preserve the *content* of the source image through cross attention tuning (CAT) for both inverse problems and image editing.

Tables 4(d) and (e) study the impact of the diffusion steps and stochastic averaging steps. While the diffusion steps seem to be the larger the better, we observe 50 DDIM steps suffice for achieving satisfactory results. For the latter, too many stochastic averaging steps result in deviation from the small neighborhood around the forward latent. Thus, we use a modest number of stochastic averaging steps (*i.e.*, 5) of proximal gradient update, as detailed in §3.2.

5.3. Results on Image Editing

Qualitative Study: STSL seamlessly extends to editing corrupted image. As shown in Figure 3, both NTI [35] and commercial software exhibit lower editing quality for real images with or without corruption. NTI [35] struggles to generate quality images with corruptions. The commercial software demands user-selected regions for editing. In contrast, STSL-CAT accurately localizes the edits without user intervention, and preserves the integrity of the entire image. For example, in NTI [35], replacing the car with a bicycle affects the surrounding house, and transforming the house to a supermarket changes the SUV into a sedan.

Quantitative Study: Table 4(f) shows a quantitative analysis of 100 randomly selected dog images from ImageNet for "a dog" to "a cat" editing. NTI-CAT incorporates NTI [35] with our Cross-Attention-Tuning (§3.3) in the reverse process, and shows consistently improvement over NTI on the clean image editing (the first two columns). The CLIP accuracy [18] remains similar, because it doesn't measure content preservation but only the matching between the output and the target prompt. In corrupted image editing (the last two columns), our end-to-end STSL-CAT pipeline demonstrates effectiveness by surpassing NTI [35] with a notable relative improvement of 32% in CLIP accuracy.

6. Conclusion

We introduced STSL, a novel posterior sampler that combines the efficiency of the first-order Tweedie with a tractable second-order approximation in a new reverse process. Our theoretical results show that our surrogate loss, requiring only an estimate of the trace of the Hessian, establishes a lower bound for the second-order approximation. STSL achieves 4X and 8X reduction in neural function evaluations compared to SoTA solvers PSLD [43] and P2L [10], respectively, while enhancing sampling quality across various inversion tasks. STSL extends to text-guided image editing, surpassing NTI [35] in handling corrupted images. To our best knowledge, this marks the first *efficient* second-order approximation for solving inverse problems using latent diffusion and image editing with corruption.

Acknowledgements: This research is supported by NSF Grant 2019844, a Google research collaboration award, and the UT Austin Machine Learning Lab. Litu Rout is supported by Ju-Nam and Pearl Chew Presidential Fellowship in Engineering and George J. Heuer Graduate Fellowship.

References

[1] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 3

[2] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold Diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022. 2

[3] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Linear convergence bounds for diffusion models via stochastic localization. *arXiv preprint arXiv:2308.03686*, 2023. 4

[4] V. Borkar. *Stochastic Approximation, A Dynamical Systems Viewpoint*. Springer, 2008. 4

[5] Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and O Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *arXiv preprint arXiv:2310.06721*, 2023. 2, 3, 5

[6] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022. 4

[7] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *Advances in Neural Information Processing Systems*, 2022. 1, 2

[8] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 3, 4, 5, 6, 7, 11, 14, 17

[9] Hyungjin Chung, Jeongsol Kim, and Jong Chul Ye. Direct diffusion bridge using data consistency for inverse problems. *arXiv preprint arXiv:2305.19809*, 2023. 2

[10] Hyungjin Chung, Jong Chul Ye, Peyman Milanfar, and Mauricio Delbracio. Prompt-tuning latent diffusion models for inverse problems. *arXiv preprint arXiv:2310.01110*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 14, 15, 16, 17, 18, 19

[11] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G Dimakis, and Peyman Milanfar. Soft diffusion: Score matching for general corruptions. *arXiv preprint arXiv:2209.05442*, 2022. 2

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 5, 7

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1

[14] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. 5, 11

[15] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, 2005. 5, 14

[16] Linchao He, Hongyu Yan, Mengting Luo, Kunming Luo, Wang Wang, Wenchao Du, Hu Chen, Hongyu Yang, and Yi Zhang. Iterative reconstruction based on latent diffusion model for sparse data reconstruction. *arXiv preprint arXiv:2307.12070*, 2023. 6, 17

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3, 4, 15

[18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 8

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3, 5

[20] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18 (3):1059–1076, 1989. 5, 14

[21] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 2, 3

[22] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34:14938–14954, 2021. 2

[23] Ajil Jalal, Sushrut Karmalkar, Alexandros G Dimakis, and Eric Price. Instance-optimal compressed sensing via posterior sampling. *arXiv preprint arXiv:2106.11438*, 2021. 2

[24] Ajil Jalal, Sushrut Karmalkar, Jessica Hoffmann, Alex Dimakis, and Eric Price. Fairness for image generation with uncertain sensitive attributes. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4721–4732. PMLR, 2021. 2

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2, 5, 7

[26] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*. 6

[27] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 2, 3

[28] Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and content representation. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3, 4, 5

[29] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. 4

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 6, 7

[31] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[32] Chenlin Meng, Yang Song, Wenzhe Li, and Stefano Ermon. Estimating high order gradients of the data distribution by denoising. *Advances in Neural Information Processing Systems*, 34:25359–25369, 2021. 1, 2, 3, 5

[33] Xiangming Meng and Yoshiyuki Kabashima. Diffusion model based posterior sampling for noisy linear inverse problems. *arXiv preprint arXiv:2211.12343*, 2022. 7

[34] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020. 2

[35] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 1, 2, 3, 4, 6, 7, 8, 15, 23, 25

[36] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013. 3

[37] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2, 3, 6

[38] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[40] H ROBBINS. An empirical bayes approach to statistics. In *Proc. 3rd Berkeley Symp. Math. Statist. Probab., 1956*, pages 157–163, 1956. 5, 11

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 3, 5, 13

[42] Litu Rout, Advait Parulekar, Constantine Caramanis, and Sanjay Shakkottai. A theoretical justification for image inpainting using denoising diffusion probabilistic models. *arXiv preprint arXiv:2302.01217*, 2023. 2, 5

[43] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G Dimakis, and Sanjay Shakkottai. Solving inverse problems provably via posterior sampling with latent diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 13, 14, 15, 16, 17, 19

[44] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 2, 3

[45] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2022. 6, 14

[46] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 3

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265, Lille, France, 2015. PMLR. 2, 3

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3, 4, 15

[49] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. 1, 2, 5, 6

[50] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[51] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. In *International Conference on Learning Representations*. 2

[52] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10748–10757, 2022. 5

[53] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 2, 3

[54] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[55] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1229, 2023. 6

Appendix

This section includes more results and details that did not fit into the main paper due to space limitation. Particularly, we offer expanded theoretical analysis in §A and implementation details in §B, along with other supportive analysis. These sections provide a deeper understanding and comprehensive context to the research presented in the main body of the paper.

A. Theoretical Analysis

A.1. Posterior mean and covariance using Tweedie's formula

Proposition A.1 ([14, 40]). *Given $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ and $\epsilon \sim \mathcal{N}(0, I)$, denote by $\bar{X}_0 = \mathbb{E}_{X_0 \sim p_t(X_0|X_t=\mathbf{x}_t)}[X_0]$ the posterior mean of $p_t(X_0|X_t = \mathbf{x}_t)$. Then, for the variance preserving SDE or DDPM sampling, $p_t(X_0|X_t = \mathbf{x}_t)$ has mean*

$$\mathbb{E}_{X_0 \sim p_t(X_0|X_t=\mathbf{x}_t)}[X_0] = \frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}} + \frac{(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\nabla_{\mathbf{x}_t}\log p_t(X_t = \mathbf{x}_t)$$

and covariance

$$\mathbb{E}_{X_0 \sim p_t(X_0|X_t=\mathbf{x}_t)}\left[\left(X_0 - \bar{X}_0\right)\left(X_0 - \bar{X}_0\right)^T\right] = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\left(\mathbf{I} + (1-\bar{\alpha}_t)\nabla^2_{\mathbf{x}_t}\log p_t(X_t = \mathbf{x}_t)\right).$$

Proof. Given $\mathbf{x}_t = \mu + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we know that $\mathbf{x}_t \sim \mathcal{N}(\mu, \sigma^2\mathbf{I})$. From [14, Section 2], we have

$$\begin{aligned}\mathbb{E}[\mu|\mathbf{x}_t] &= \mathbf{x}_t + \sigma^2\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t),\\ \mathbb{V}[\mu|\mathbf{x}_t] &= \sigma^2\left(1+\sigma^2\right)\nabla^2_{\mathbf{x}_t}\log p_t(\mathbf{x}_t),\end{aligned}$$

where $\mathbb{E}[\mu|\mathbf{x}_t]$ and $\mathbb{V}[\mu|\mathbf{x}_t]$ denote the conditional mean and the conditional variance, respectively. Since $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ in our case, we get

$$\begin{aligned}\mathbb{E}\left[\sqrt{\bar{\alpha}_t}\mathbf{x}_0|\mathbf{x}_t\right] &= \sqrt{\bar{\alpha}_t}\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \mathbf{x}_t + (1-\bar{\alpha}_t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t),\\ \mathbb{V}\left[\sqrt{\bar{\alpha}_t}\mathbf{x}_0|\mathbf{x}_t\right] &= \bar{\alpha}_t\mathbb{V}[\mathbf{x}_0|\mathbf{x}_t] = (1-\bar{\alpha}_t)(1+(1-\bar{\alpha}_t))\nabla^2_{\mathbf{x}_t}\log p_t(\mathbf{x}_t),\end{aligned}$$

which upon rearrangement yields the following:

$$\begin{aligned}\mathbb{E}_{X_0 \sim p_t(X_0|X_t=\mathbf{x}_t)}[X_0] &= \frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}} + \frac{(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\nabla_{\mathbf{x}_t}\log p_t(X_t = \mathbf{x}_t)\\ \mathbb{E}_{X_0 \sim p_t(X_0|X_t=\mathbf{x}_t)}\left[\left(X_0 - \bar{X}_0\right)\left(X_0 - \bar{X}_0\right)^T\right] &= \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\left(\mathbf{I} + (1-\bar{\alpha}_t)\nabla^2_{\mathbf{x}_t}\log p_t(X_t = \mathbf{x}_t)\right).\end{aligned}$$

This completes the proof of the statement. □

A.2. First-order Tweedie sampler

Theorem A.2. *(First-order Tweedie Estimator [8]). Given measurements $\mathbf{y} = \mathcal{A}(\mathbf{z}_T) + \mathbf{n}$, $\mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2_{\mathbf{y}}\mathbf{I}\right)$ and the first-order approximation $p_{T-t}(\mathbf{y}|Z_t) \approx p_{T-t}(\mathbf{y}|\bar{Z}_T)$, define the Jensen's gap as:*

$$\mathcal{J} := \left|\mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t)}[p_{T-t}(\mathbf{y}|Z_t)] - p_{T-t}(\mathbf{y}|\bar{Z}_T)\right|,$$

where $\bar{Z}_T := \mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t)}[Z_T]$. Then, the error due to first-order approximation is upper bounded by

$$\mathcal{J} \leq \frac{d}{\sqrt{2\pi\sigma^2_{\mathbf{y}}}}\exp\left(-\frac{1}{2\sigma^2_{\mathbf{y}}}\right)\|\nabla_z\mathcal{A}(\mathbf{z})\|m_1,$$

where $\|\nabla_{\mathbf{z}}\mathcal{A}(\mathbf{z})\| := \max_{\mathbf{z}}\|\mathcal{A}(\mathbf{z})\|$ and $m_1 := \int \|Z_T - \bar{Z}_T\|\, p_{T-t}(Z_T|Z_t)dZ_T$.

Since $\|\nabla_{\mathbf{z}}\mathcal{A}(\mathbf{z})\|$ and m_1 are finite for most inverse problems, the Jensen's gap goes to zero as $\sigma_{\mathbf{y}} \to \infty$, leading to less approximation error in (2). This setting is of less practical significance because as $\sigma_{\mathbf{y}} \to \infty$, the measurements $\mathbf{y} = \mathcal{A}(\mathbf{z}_T) + \sigma_{\mathbf{y}}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ provide no meaningful information about $\mathbf{z}_T$. Thus, sampling from the posterior $p_0(Z_T|\mathbf{y}) = p_0(X_0|\mathbf{y})$ is as good as sampling from the prior $p_0(X_0)$. On the other hand, when $\sigma_{\mathbf{y}} \to 0$, the problem is reduced to a noiseless setting which is relatively easier to deal with. In practically relevant settings where σ_y is non-zero and finite, the Jensen's gap could be arbitrarily large. This leads to a bias in reconstruction and sub-optimal performance in various tasks as we show in §5.

A.3. Second-order Tweedie sampler from surrogate loss

Theorem A.3 (Tweedie Sampler from Surrogate Loss)**.** *Suppose **Assumption** 4.2 and **Assumption** 4.3 hold. Let $\hat{\mathcal{L}}(\mathbf{y}, Z_t)$ denote the function:*

$$\hat{\mathcal{L}}(\mathbf{y}, Z_t) := \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(1 - \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}md - (1-\bar{\alpha}_t)m\,Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right).$$

For $\lambda = \mathcal{O}(\frac{1}{\sigma_{\mathbf{y}}^2})$ and $\gamma = \mathcal{O}(\frac{\eta}{d})$, the following holds: $\hat{\mathcal{L}}(\mathbf{y}, Z_t) \leq \log p_{T-t}\left(\mathbf{y}|Z_t\right)$. Further, the gradient of $\hat{\mathcal{L}}(\mathbf{y}, Z_t)$ is given by:

$$\nabla\hat{\mathcal{L}}(\mathbf{y}, Z_t) = -\frac{1}{2\sigma_{\mathbf{y}}^2}\nabla\|\mathbf{y}-\mathbf{A}\bar{Z}_T\|^2 - \frac{(1-\bar{\alpha}_t)m}{\left(1 - \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}md - (1-\bar{\alpha}_t)m\,Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right)}\nabla Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right).$$

Proof. We want to compute the following:

$$\begin{aligned}
\log p_{T-t}(\mathbf{y}|Z_t) &= \log \int p_{T-t}(\mathbf{y}|Z_t, Z_T)p_{T-t}(Z_T|Z_t)dZ_T \\
&\overset{(i)}{=} \log \int p_{T-t}(\mathbf{y}|Z_T)p_{T-t}(Z_T|Z_t)dZ_T \\
&= \log \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[p_{T-t}(\mathbf{y}|Z_T)\right] \qquad (7)
\end{aligned}$$

where (i) is because $\mathbf{y}$ is independent of Z_t given Z_T. Denote by $\bar{Z}_T = \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[Z_T\right]$. Now, using Taylor series expansion at $\bar{Z}_T$, for some $\tilde{Z}_T \in \mathcal{B}_r\left(\bar{Z}_T\right) := \{Z \in \mathbb{R}^d | \|Z - \bar{Z}_T\| \leq r\}, r = \|Z_T - \bar{Z}_T\|$, we get

$$\begin{aligned}
&\log \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[p_{T-t}(\mathbf{y}|Z_T)\right] \\
&= \log \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[p_{T-t}(\mathbf{y}|\bar{Z}_T) + \langle \nabla p_{T-t}(\mathbf{y}|Z_t)\,|_{\bar{Z}_T}, Z_T - \bar{Z}_T\rangle + \frac{1}{2}(Z_T - \bar{Z}_T)^T\nabla^2 p_{T-t}(\mathbf{y}|\tilde{Z}_T)(Z_T - \bar{Z}_T)\right] \\
&= \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T) + \frac{1}{2}\mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[(Z_T - \bar{Z}_T)^T\nabla^2 p_{T-t}(\mathbf{y}|\tilde{Z}_T)(Z_T - \bar{Z}_T)\right]\right),
\end{aligned}$$

where the last step follows from linearity of expectation and the fact that $\langle \nabla p_{T-t}(\mathbf{y}|Z_t)\,|_{\bar{Z}_T}, \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[Z_T\right] - \bar{Z}_T\rangle = 0$. Since $\log(a+b) = \log(a) + \log(1 + b/a)$ for $a > 0$ and $p_{T-t}(\mathbf{y}|\bar{Z}_T) > 0$ due to **Assumption** 4.2, the above expression simplifies to

$$\begin{aligned}
&\log \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[p_{T-t}(\mathbf{y}|Z_T)\right] \\
&= \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(1 + \frac{\mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[(Z_T - \bar{Z}_T)^T\nabla^2 p_{T-t}(\mathbf{y}|\tilde{Z}_T)(Z_T - \bar{Z}_T)\right]}{2p_{T-t}(\mathbf{y}|\bar{Z}_T)}\right) \\
&= \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(1 + \mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[(Z_T - \bar{Z}_T)^T\left(\frac{\nabla^2 p_{T-t}(\mathbf{y}|\tilde{Z}_T)}{2p_{T-t}(\mathbf{y}|\bar{Z}_T)}\right)(Z_T - \bar{Z}_T)\right]\right) \\
&\geq \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(1 - m\,\mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[(Z_T - \bar{Z}_T)^T(Z_T - \bar{Z}_T)\right]\right) \\
&= \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(1 - m\,Trace\left(\mathbb{E}_{Z_T\sim p_{T-t}(Z_T|Z_t)}\left[\left(Z_T - \bar{Z}_T\right)\left(Z_T - \bar{Z}_T\right)^T\right]\right)\right) \\
&= \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(1 - m\,Trace\left(\frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}}\left(I + (1-\bar{\alpha}_{T-t})\nabla^2 \log p_{T-t}(Z_t)\right)\right)\right) \\
&= \log\left(p_{T-t}(\mathbf{y}|\bar{Z}_T)\right) + \log\left(1 - \frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}}md - (1-\bar{\alpha}_{T-t})m\,Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right) := \hat{\mathcal{L}}(\mathbf{y}, Z_t)
\end{aligned}$$

This completes the proof of the first part, $\hat{\mathcal{L}}(\mathbf{y}, Z_t) \leq \log p_{T-t}\left(\mathbf{y}|Z_t\right)$.

Next, the gradient of the lower bound with respect to Z_t becomes:

$$
\begin{aligned}
&\nabla \hat{\mathcal{L}}(\mathbf{y}, Z_t) \\
&= -\frac{1}{2\sigma_{\mathbf{y}}^2}\nabla\|\mathbf{y} - \mathbf{A}\bar{Z}_T\|^2 + \nabla \log\left(1 - \frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}} md - (1-\bar{\alpha}_{T-t})m\, Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right) \\
&= -\frac{1}{2\sigma_{\mathbf{y}}^2}\nabla\|\mathbf{y} - \mathbf{A}\bar{Z}_T\|^2 - \frac{(1-\bar{\alpha}_{T-t})m}{\left(1 - \frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}} md - (1-\bar{\alpha}_{T-t})m\, Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right)} \nabla Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right),
\end{aligned}
$$

where the last step follows from $\nabla\left(1 - \frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}} md\right) = 0$. □

Implication: From the above result, we have

$$
\nabla \hat{\mathcal{L}}(\mathbf{y}, Z_t) \simeq -\lambda \nabla\|\mathbf{y} - \mathbf{A}\bar{Z}_T\|^2 - \gamma \nabla\left(Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\right),
$$

where $\lambda = \mathcal{O}\left(\frac{1}{\sigma_{\mathbf{y}}^2}\right)$ and $\gamma = \mathcal{O}\left(\frac{\eta}{d}\right)$ are hyper-parameters to be tuned in practice.

Connection with the surrogate loss: The gradient of the lower bound $\hat{\mathcal{L}}(\mathbf{y}, Z_t)$ is equal to the negative gradient of the surrogate loss function $\mathcal{L}(\mathbf{y}, Z_t)$ introduced in §3.2 and §4, i.e., $\nabla \hat{\mathcal{L}}(\mathbf{y}, Z_t) \simeq -\nabla \mathcal{L}(\mathbf{y}, Z_t)$, when the constants λ and γ are chosen appropriately. More precisely, as given in the statement of the **Theorem A.3**, these gradients are equal when $\lambda = \frac{-1}{2\sigma_{\mathbf{y}}^2}$ and $\gamma = \frac{-(1-\bar{\alpha}_{T-t})m}{\left(1 - \frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}} md - (1-\bar{\alpha}_{T-t})m\, Trace(\nabla^2 \log p_{T-t}(Z_t))\right)}$. In our implementation, we use $\nabla \mathcal{L}(\mathbf{y}, Z_t)$ that results in proximal gradient *descent* in **Algorithm 1**.

Remark A.4 (Second-order Tweedie for Gaussian Prior)**.** *Recall from Appendix A.3 that we want to compute*

$$
\log p_{T-t}(\mathbf{y}|Z_t) = \log \mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t)}\left[p_{T-t}(\mathbf{y}|Z_T)\right].
$$

Let us suppose that the prior is Gaussian, i.e., $p_T(Z_T) = \mathcal{N}(Z_T; \boldsymbol{\mu}, \mathbf{I})$. Then, the forward and reverse process become Gaussian processes. Therefore, we can compute the posterior mean and covariance analytically using Proposition A.1 as:

$$
\begin{aligned}
\mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t=\mathbf{z}_t)}\left[Z_T\right] &= \frac{\mathbf{z}_t}{\sqrt{\bar{\alpha}_{T-t}}} + \frac{(1-\bar{\alpha}_{T-t})}{\sqrt{\bar{\alpha}_{T-t}}} \nabla_{\mathbf{z}_t} \log p_{T-t}(Z_t = \mathbf{z}_t) \\
&= \frac{\mathbf{z}_t}{\sqrt{\bar{\alpha}_{T-t}}} + \frac{(1-\bar{\alpha}_{T-t})}{\sqrt{\bar{\alpha}_{T-t}}}\left(\sqrt{\bar{\alpha}_{T-t}}\boldsymbol{\mu} - \mathbf{z}_t\right) \\
&= \sqrt{\bar{\alpha}_{T-t}}\mathbf{z}_t + (1-\bar{\alpha}_{T-t})\boldsymbol{\mu}, \qquad (8)
\end{aligned}
$$

$$
\mathbb{E}_{Z_T \sim p_{T-t}(Z_T|Z_t=\mathbf{z}_t)}\left[\left(Z_T - \bar{Z}_T\right)\left(Z_T - \bar{Z}_T\right)^T\right] = \frac{1-\bar{\alpha}_{T-t}}{\bar{\alpha}_{T-t}}\left(\mathbf{I} + (1-\bar{\alpha}_{T-t})\nabla^2_{\mathbf{z}_t} \log p_{T-t}(Z_t = \mathbf{z}_t)\right) = (1-\bar{\alpha}_{T-t})\,\mathbf{I}. \qquad (9)
$$

Thus, we obtain $p_{T-t}\left(Z_T|Z_t = \mathbf{z}_t\right) = \mathcal{N}\left(Z_T; \sqrt{\bar{\alpha}_{T-t}}\mathbf{z}_t + (1-\bar{\alpha}_{T-t})\,\boldsymbol{\mu}, (1-\bar{\alpha}_{T-t})\,\mathbf{I}\right)$ [4]. Following similar arguments from the proof in Appendix A.3, if we truncate $p_{T-t}\left(\mathbf{y}|Z_T\right)$ up to second-order terms in Taylor's expansion, then the lower bound only has an additive error by appropriately chosen stepsize. Hence, the gradients match up to some scaling factor.

Note that our theoretical analysis is provided for pixel-space diffusion models. However, it easily extends to latent diffusion models using proof techniques from PSLD [43]. Importantly, the latent space of latent diffusion models, such as Stable Diffusion [41] is usually Gaussian, which makes STSL a reasonable algorithm in practice.

[4] Instead of expanding the term inside expectation as in Appendix A.3, we can exactly compute $p_{T-t}\left(\mathbf{y}|Z_t = \mathbf{z}_t\right)$ by its second-order Taylor's expansion around the posterior mean. Therefore, for a Gaussian prior, this second-order approximation is exact. However, a similar treatment requires Hessian for non-Gaussian prior, which is computationally expensive in practice.

A.4. Computation using Hutchinson's Trace Estimator

Given $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the trace of the Hessian can be efficiently computed as:

$$\mathbb{E}\left[\epsilon^T\left(\nabla \log p_{T-t}\left(Z_t+\epsilon\right)-\nabla \log p_{T-t}\left(Z_t\right)\right)\right]-\mathcal{O}(\|\epsilon\|^3) \simeq Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right).$$

To see this, for $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, using Taylor series expansion of the score, we get

$$\nabla \log p_{T-t}\left(Z_t+\epsilon\right) \simeq \nabla \log p_{T-t}\left(Z_t\right)+\nabla^2 \log p_{T-t}(Z_t)\epsilon+\mathcal{O}\left(\|\epsilon\|^2\right).$$

Subtracting $\nabla \log p_{T-t}\left(Z_t\right)$ from both sides, and taking projection onto ϵ, we have

$$\epsilon^T\left(\nabla \log p_{T-t}\left(Z_t+\epsilon\right)-\nabla \log p_{T-t}\left(Z_t\right)\right) \simeq \epsilon^T \nabla^2 \log p_{T-t}(Z_t)\epsilon+\mathcal{O}(\|\epsilon\|^3).$$

The claim follows by taking the expectation of both sides and applying Hutchinson's trace estimator [20] as given below:

$$\begin{aligned}\mathbb{E}\left[\epsilon^T\left(\nabla \log p_{T-t}\left(Z_t+\epsilon\right)-\nabla \log p_{T-t}\left(Z_t\right)\right)\right]-\mathcal{O}(\|\epsilon\|^3) &\simeq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\left(\epsilon^T \nabla^2 \log p_{T-t}(Z_t)\epsilon\right)\right] \\ &= Trace\left(\nabla^2 \log p_{T-t}(Z_t)\right)\end{aligned}$$

The last step above involves an approximation of a higher derivative through an expectation of random projections of perturbed function evaluations. This approach has been well studied in online learning settings and with formal guarantees (e.g., Lemma 2.1 in [15]). In our case, the approximation additionally involves a "centering" with $\epsilon^T \nabla \log p_{T-t}\left(Z_t\right)$. While this terms is zero in expectation, it is useful to keep because as we discuss in Section 3.2, we are evaluating the expectation through stochastic averaging with finitely many steps. This centering decreases the magnitude of each step, thus resulting in variance improvement (and thus a less noisy approximation with a fewer number of steps).

B. Additional Experimental Evaluation

B.1. Implementation Details

Image Inversion: We follow the same experimental setup as prior works [8, 43], and use the measurement operators provided in their original source code: DPS[5] and PSLD[6]. We employ a Gaussian blur kernel (size 61×61, $\sigma = 3.0$) for *Gaussian deblurring* and a motion blur kernel (size 61×61, intensity 0.5) for *motion deblurring* tasks. For *super-resolution*, we use $4\times$ and $8\times$ downsampling as measurement operator. Additionally, we introduce 2% salt and pepper noise for *denoising* and 40% drop rate for *random inpainting* tasks. For *free-form inpainting*, we adopt the 10%-20% damage range as utilized in prior works [10, 45].

Our refinement module in **Algorithm 1** uses the Adam optimizer, with an initial learning rate of $1e-2$ and decrementing by a factor of 0.998 per diffusion time step. This process optimizes the latents with stochastic averaging. Notably, STSL exhibits robustness across various tasks, showing minimal sensitivity to hyper-parameter changes. Therefore, we maintain consistent configurations for all tasks, where $N = 2$, $\eta = 0.02$, $\nu = 2$ and $\lambda = 1$. We use $K = 5$ and $T = 50$ as defualt and conduct extensive ablation studies for free-form image inpainting task in §B.4. Following the experimental setting of P2L[10], we add independent and identically distributed Gaussian noise $\mathcal{N}\left(\mathbf{0}, 0.01^2\right)$ to each pixel.

Image Editing: In image editing, we use a single stochastic averaging step $K = 1$ since the latents have been refined during proximal gradient updates. We use $\nu = 0.02$ for the contrastive loss without normalization by the data dimension d, $\lambda = 1$ for the measurement loss and the same coefficient for Hutchinson's trace estimator $\eta = 0.02$ as in inversion. More details are elaborated in **Algorithm 2**. For the qualitative demonstration, we compare with NTI[7] and a commercial platform that is publicly available. We conduct the experiments using the latest version of the commercial software by November 2023.

Reproducibility: The pseudo-code of STSL for inverse is given in **Algorithm 1** and editing in **Algorithm 2**. All the hyper-parameter details are provided in §5 and §B.1.

B.2. Computational Complexity

Table 2 provides a comparative analysis of the runtime performance across various state-of-the-art methods. NFEs are computed based on the required reverse and optimization steps. For instance, P2L [10] demands 1000 reverse steps, accompanied

[5] https://github.com/DPS2022/diffusion-posterior-sampling
[6] https://github.com/LituRout/PSLD
[7] https://github.com/google/prompt-to-prompt/

Algorithm 2: Second-order Tweedie sampler from Surrogate Loss (STSL) for image inversion and editing task

Input: Diffusion time steps T, observed $\mathbf{y}$, measurement operator $\mathbf{A}$, encoder $\mathcal{E}$, decoder $\mathcal{D}$, learned score $\boldsymbol{s}_\theta$, target text "*prompt*", text encoder Φ

Tunable Parameters: likelihood strength λ, stochastic averaging steps K, second-order correction stepsize η,

Output: Edited Image $\mathcal{D}(Z_T)$

Initialization: $\overrightarrow{Z}_0 = \mathcal{E}(\mathbf{A}^T\mathbf{y})$ ▷ DDIM forward process [35, 48]

for $t = 0$ **to** $T-1$ **do**

  $\overrightarrow{Z}_{t+1} \leftarrow \sqrt{\frac{\bar{\alpha}_{t+1}}{\bar{\alpha}_t}}\overrightarrow{Z}_t - \left(\sqrt{\frac{1}{\bar{\alpha}_{t+1}}-1} - \sqrt{\frac{1}{\bar{\alpha}_t}-1}\right)\left(\sqrt{1-\bar{\alpha}_t}\right) s_\theta(\overrightarrow{Z}_t, t)$

end

Initialization: $Z_0 = \overrightarrow{Z}_T$ ▷ proposed reverse process for image inversion

for $t = 0$ **to** $T-1$ **do**

  for $k = 0$ **to** K **do**

    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ stochastic averaging

    $\bar{Z}_T \leftarrow (Z_t + (1-\bar{\alpha}_{T-t})\boldsymbol{s}_\theta(Z_t, T-t))/\sqrt{\bar{\alpha}_{T-t}}$

    $Z_t \leftarrow Z_t - \lambda\nabla\|\mathbf{y} - \mathbf{A}\mathcal{D}(\bar{Z}_T)\| - (\eta/d)\nabla\left(\epsilon^T\boldsymbol{s}_\theta\left(Z_t+\epsilon, T-t\right) - \epsilon^T\boldsymbol{s}_\theta\left(Z_t, T-t\right)\right)$

  end

  $\bar{Z}_T \leftarrow (Z_t + (1-\bar{\alpha}_{T-t})\boldsymbol{s}_\theta(Z_t, T-t))/\sqrt{\bar{\alpha}_{T-t}}$

  $Z_{t+1} \leftarrow \frac{\sqrt{\alpha_{T-t}}(1-\bar{\alpha}_{T-t-1})}{1-\bar{\alpha}_{T-t}}Z_t + \frac{\sqrt{\bar{\alpha}_{T-t-1}}(1-\alpha_{T-t})}{1-\bar{\alpha}_{T-t}}\bar{Z}_T$

end

Initialization: $Z_0 = \overrightarrow{Z}_T$ ▷ proposed reverse process for image editing

for $t = 0$ **to** $T-1$ **do**

  $\bar{Z}_T \leftarrow (Z_t + (1-\bar{\alpha}_{T-t})\boldsymbol{s}_\theta(Z_t, T-t, \varphi_t))/\sqrt{\bar{\alpha}_{T-t}}$

  $f\left(Z_t, T-t, \varphi_t\right) = \sqrt{\bar{\alpha}_{T-t-1}}\bar{Z}_T + \sqrt{1-\bar{\alpha}_{T-t-1}}\sqrt{1-\bar{\alpha}_{T-t}}\boldsymbol{s}_\theta(Z_t, T-t, \varphi_t)$

  $\hat{\varphi}_t = \arg\min_{\varphi_t}\|Z_{t+1} - f\left(Z_t, T-t, \varphi_t\right)\|_2^2$ ▷ Null-optimization

  $\hat{Z}_{t+1} \leftarrow CAC(Z_t, T-t, \hat{\varphi}_t, \Phi\{\text{"}prompt\text{"}\})$ ▷ Cross-Attention-Control (CAC) [17]

  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

  $\bar{Z}_T \leftarrow (Z_t + (1-\bar{\alpha}_{T-t})\boldsymbol{s}_\theta(Z_t, T-t, \Phi\{\text{"}prompt\text{"}\}))/\sqrt{\bar{\alpha}_{T-t}}$

  $Z_{t+1} \leftarrow \hat{Z}_{t+1} - \lambda\nabla\|\mathbf{y} - \mathbf{A}\mathcal{D}(\bar{Z}_T)\| - \frac{\eta}{d}\nabla\epsilon^T\left(\boldsymbol{s}_\theta\left(Z_t+\epsilon, T-t, \Phi\{\text{"}prompt\text{"}\}\right) - \boldsymbol{s}_\theta\left(Z_t, T-t, \Phi\{\text{"}prompt\text{"}\}\right)\right)$

end

return $\mathcal{D}(Z_T)$

by at least one prompt tuning step per iteration, accumulating in a total of 2000 NFEs. The best results of P2L [10] are obtained with around 5000 NFEs, which amounts to **30 mins** of runtime per image. Other baseline methods require 1000 reverse steps. The best results of PSLD/GML-DPS [43] are obtained with 1000 NFEs, which amounts to **12 mins** of runtime per image. Our STSL framework demonstrates *efficiency* by employing only 50 DDIM steps coupled with 5 stochastic averaging steps, resulting in a considerably lower count of 250 NFEs. This translates into significantly lower runtime of **under 3 min** with a considerable gain in performance. Note that the runtime of PDM-solvers is lower because the underlying generative model is smaller compared to large-scale foundation models, such as Stable Diffusion. Despite smaller runtime, PDM-solvers are subpar SoTA solvers [10, 43] leveraging these foundation models.

B.3. More Qualitative Results

We present extended results of the proposed method and compare with SoTA solvers in motion deblurring (Figure 4), SRx8 (Figure 5), and Gaussian deblurring (Figure 6). Notably, STSL demonstrates superior capability in preserving intricate image details and reducing artifact generation, particularly in text-rich images. This is exemplified in the last images of Figures 4 and 5, where text clarity and legibility are visibly enhanced. Furthermore, unlike other methods that tend to introduce spurious textures, our approach consistently maintains high image fidelity, reinforcing the effectiveness of STSL in complex scenarios.

Our results also showcase the adaptability of STSL in image editing tasks. In Figures 9 and 10, we illustrate that conventional editing methods struggle with corrupted input images, whereas STSL-CAT achieves high-fidelity editing under these conditions. Furthermore, STSL-CAT excels in maintaining the integrity of the image even when the input is not corrupted,

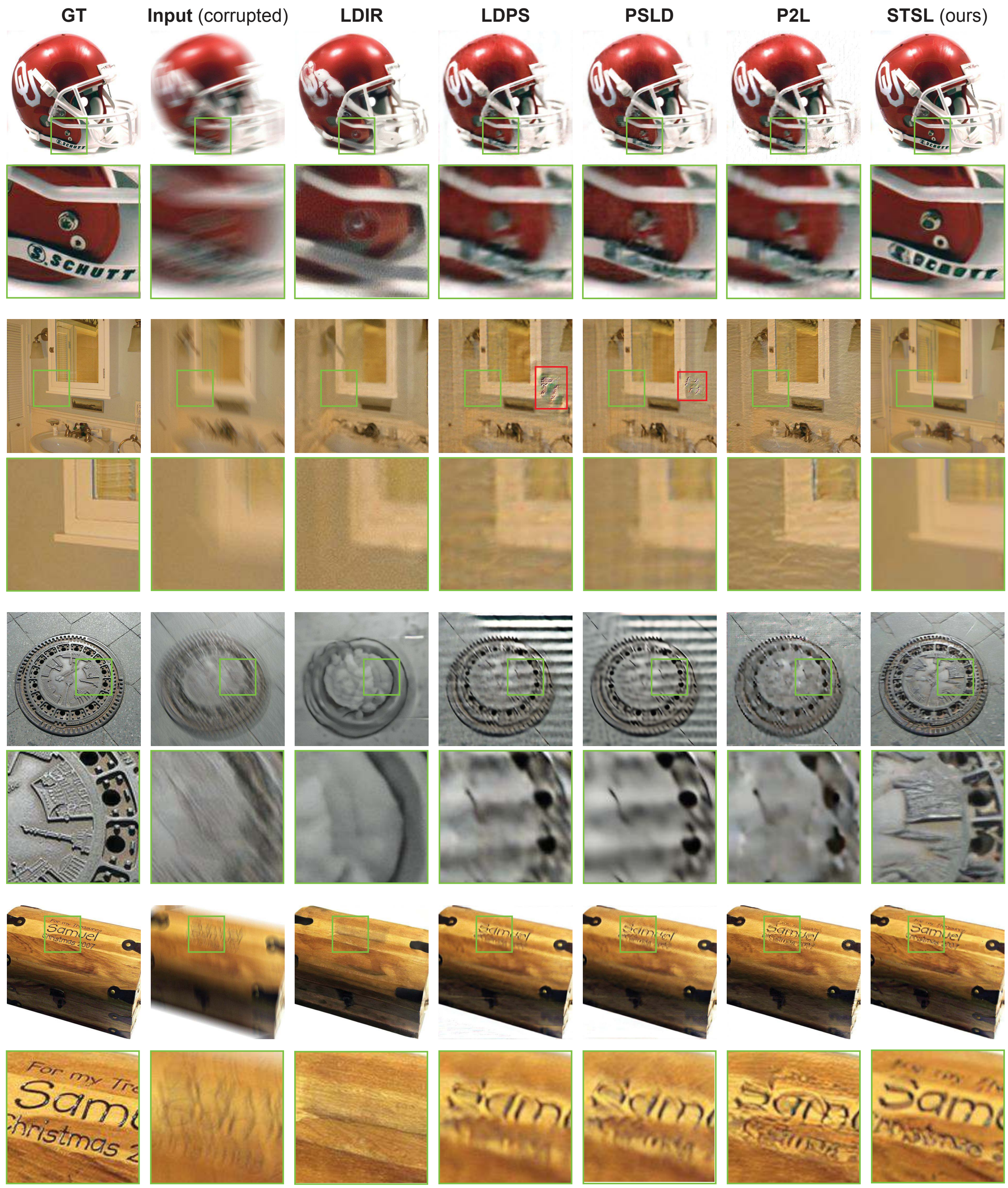

Figure 4. **Qualitative results on motion deblurring:** Odd rows represent the full image, while even rows show a zoomed-in view of the green box. The red boxes indicate artifacts from various methods. STSL demonstrates superior performance in preserving image details while simultaneously minimizing artifacts and fake textures. The competitive baselines: PSLD [43] and P2L[10] introduce artifacts and fake texture that might be mistaken as sharpness of the reconstructed image. Observe the high fidelity text restoration by the proposed approach STSL in the last row.

Method	LPIPS↓	PSNR↑	SSIM↑	K	T	NFEs	Initialization
STSL-I (Ours)	0.279	30.61	81.53	5	50	250	Alg. 1 Line 5
STSL-III (Ours)	0.282	30.79	82.55	5	200	1000	Alg. 1 Line 5
STSL-II (Ours)	0.386	29.65	77.16	5	50	250	Gaussian
STSL-IV (Ours)	<u>0.311</u>	30.29	81.74	5	200	1000	Gaussian
STSL-V (Ours)	**0.291**	30.65	82.48	2	1000	2000	Gaussian
P2L [10]	0.321	31.29	**85.16**	2	1000	2000	Gaussian
PSLD [43]	0.344	**31.54**	84.20	1	1000	1000	Gaussian
GML-DPS [43]	0.364	<u>31.49</u>	84.00	1	1000	1000	Gaussian
LDPS [43]	0.379	31.34	84.45	1	1000	1000	Gaussian
LDIR [16]	0.386	31.24	<u>84.87</u>	1	1000	1000	Gaussian
DPS [8]	0.368	28.96	69.89	1	1000	1000	Gaussian

Table 5. **Quantitative results of the free-form inpainting task on ImageNet-1K**. STSL-I/III are initialized from the forward latent $Z_0 \sim p_T(Z_0|\mathcal{E}(\mathbf{A}^T\mathbf{y}))$ while all the other methods are initialized with Gaussian noise $Z_0 \sim \pi_d$. As discussed in §B.4, STSL-I/III sometimes leaves small missing areas as shown in Figure 8 even though it better reconstructs unmasked regions of the image. To make a fair comparison, we only consider the methods using the same initialization from the Gaussian noise that successfully inpaint all the missing regions. In this setting, STSL-IV and STSL-V still outperform SoTA solver PSLD [43] and P2L [10] using the same number of NFEs: 1000 and 2000, respectively.

as demonstrated in Figure 11. These qualitative results, supporting the quantitative data presented in Table 4(f), reveal that the integration of CAT with NTI preserve image *content*, such as in areas of the nose and eyes while changing the *style*. The *refinement* of forward latents (§3.2) further contributes to this improvement in rendering details from the corrupt images.

B.4. Free-form Inpainting

The main body of the paper contains quantitative results on standard datasets. In this section, we provide additional quantitative results on free-form inpainting [10], which targets to *generate* missing pixels in the blank areas as opposed to *restore* corrupted pixels. Following prior works [10, 43], we initialize the reverse process at $Z_0 \sim \pi_d$ in STSL-II/IV/V. STSL-I/III are initialized at the forward latent $Z_0 \sim p_T(Z_0|\mathcal{E}(\mathbf{A}^T\mathbf{y}))$. Table 6 and Table 5 show the quantitative evaluation on FFHQ (512×512) and ImageNet (512×512), respectively.

We conduct ablation studies to analyze the latency-optimization trade-offs. As shown in Table 5, STSL uses different combinations of stochastic averaging steps K and DDIM steps T. When compared with methods with the same number of NFEs, STSL outperforms the SoTA solvers PSLD [43] and P2L [10] in terms of LPIPS and achieves comparable results in terms of PSNR/SSIM. Figure 7 illustrates the qualitative results on ImageNet.

Method	LPIPS	PSNR	SSIM
STSL (ours)	**0.260**	31.30	87.56
P2L [10]	0.273	32.44	**91.00**
PSLD [43]	0.312	32.42	88.58
GML-DPS [43]	0.335	**32.45**	88.66
LDPS [43]	0.372	32.12	88.15
LDIR [16]	0.338	32.25	90.28

Table 6. Additional quantitative results on FFHQ-1K.

Limitation: Figure 8 shows the failure cases of our proposed inverse problem solver STSL in free-form inpainting. We observe that the large blocks of missing pixels are embedded into the forward latents in STSL-I/III , which is hard to refine using proximal gradient updates. Therefore, the masked regions of the final reconstruction sometimes contain incomplete pixels. This issue arises due to imperfect encoder-decoder of the Stable Diffusion foundation model [43], and could be partly circumvented by slowing down the diffusion process to $T = 1000$ steps and initializing the reverse process at $Z_0 \sim \pi_d$ as in PSLD [43] and P2L [10]. We recommend following this recipe for free-form inpainting.

The proposed inverse problem solver uses $\mathbf{A}^T$ from DPS [8], which is set to identity for some tasks. It might be better to use Jax implementation of A^T for improved performance as in P2L [10].

Future work: Our approach does not tune the prompt used in the generative foundation model. Integrating prompt-tuning [10] into our pipeline might prove beneficial.

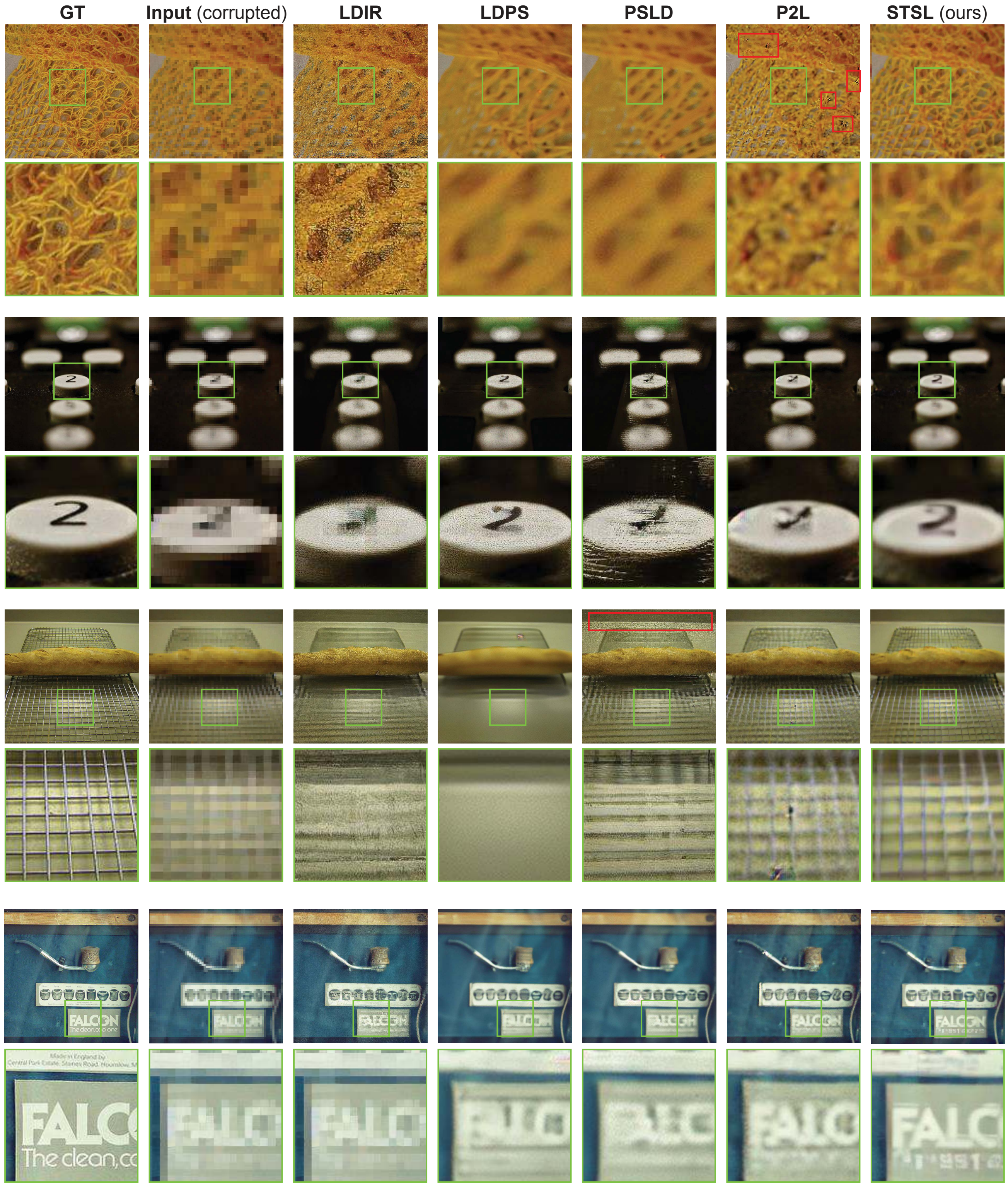

Figure 5. **Qualitative results on SRx8:** Odd rows represent the full image, while even rows show a zoomed-in view of the green box. The red boxes indicate artifacts from various methods. STSL restores image details without introducing artifacts (row 1) and shows its potentiality in restoring images with complicated patterns (row 2 and row 6). The competitive baselines: PSLD [43] and P2L [10] suffer from artifacts that are clearly visible in the highlighted regions.

Figure 6. **Qualitative results on Gaussian deblurring:** Odd rows represent the full image, while even rows show a zoomed-in view of the green box. The red boxes indicate artifacts from various methods. Row 4 and row 8 demonstrate the superior performance of STSL in restoring text and preserving details.

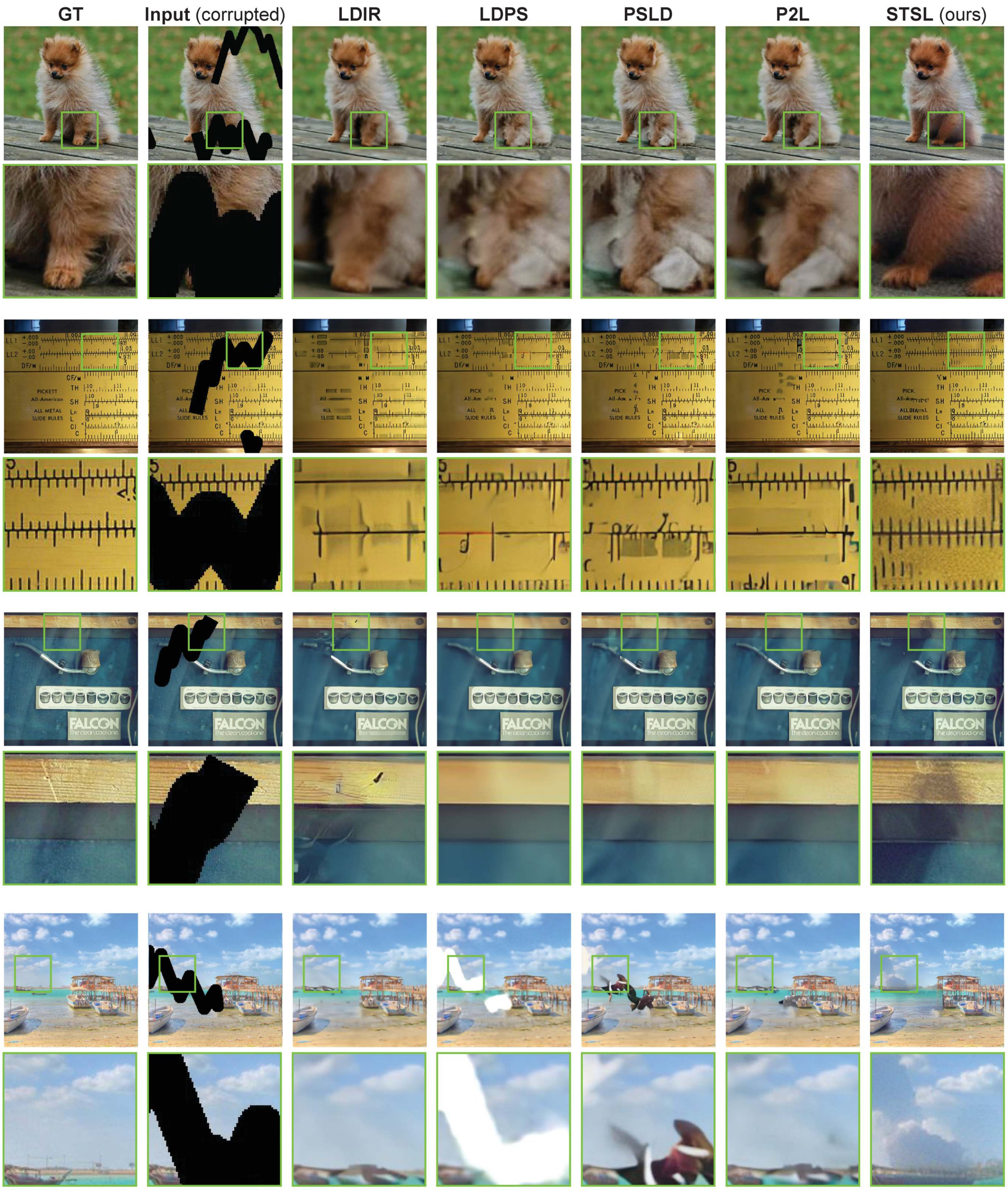

Figure 7. **Qualitative results on free-form inpainting:** Odd rows represent the full image, while even rows show a zoomed-in view of the green box. Note that the model is expected to generate new content that harmonizes with the rest of the pixels, but not necessarily reproduce the same image. This is because the goal is to sample the posterior $p(X|\mathbf{y})$. The outputs from STSL contain more detailed patterns (row 6) and clear edges (row 2&4).

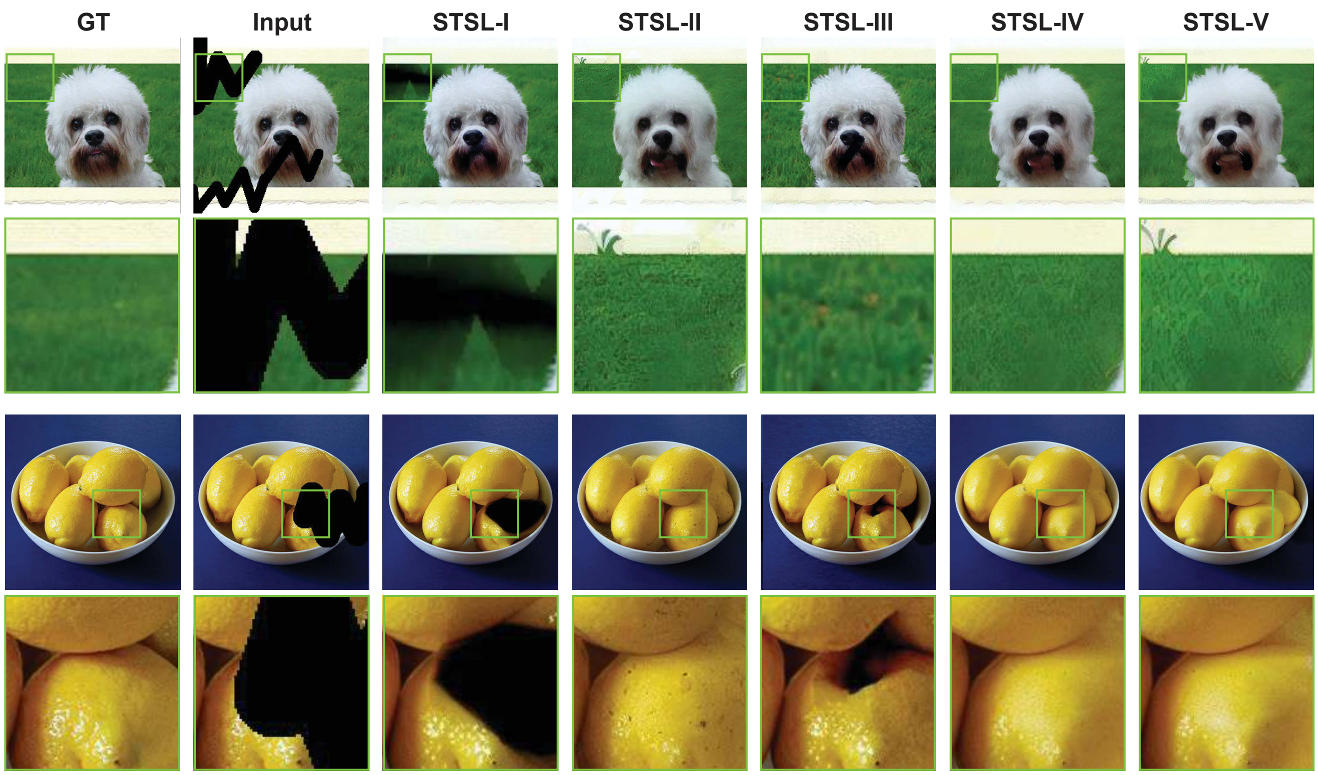

Figure 8. **Failure cases of free-form inpainting:** The restored images appear sharp when initialized with the forward latents $Z_0 \sim p_T(Z_0|\mathcal{E}(\mathbf{A}^T\mathbf{y}))$ in STSL-I/III, while the images with the reverse process initialized at $Z_0 \sim \pi_d$ yield more complete inpainting results (STSL-II/IV/V). One may choose the initialization and the corresponding hyper-parameters as per the requirement in practice.

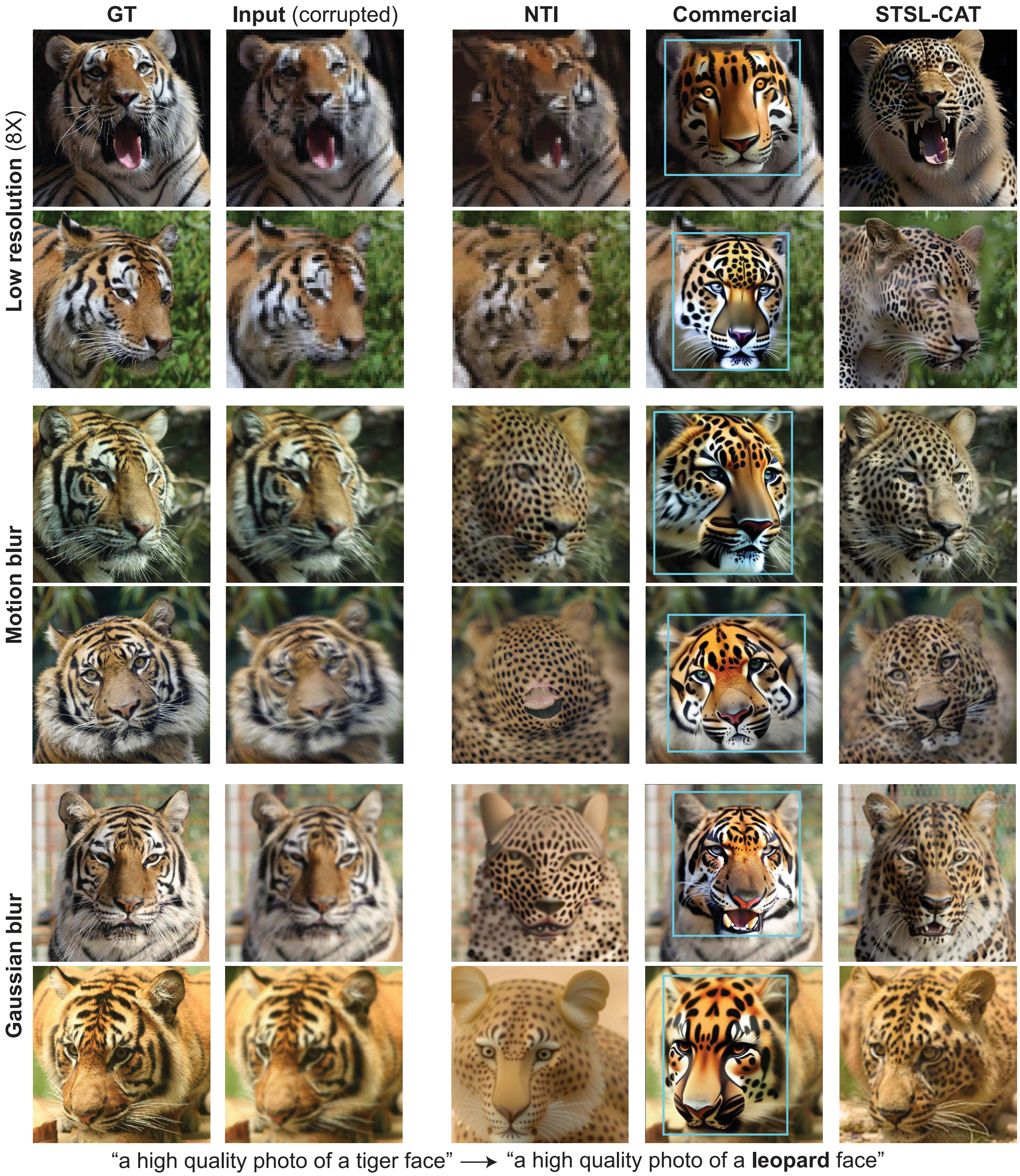

Figure 9. **Qualitative results on image editing on the corrupted images “tiger” to “leopard”.** While NTI[35] fails to conduct high-fidelity image editing when various corruptions are presented, the commercial software synthesizes artistic visual objects without preserving the content of the source image. Furthermore, the proposed method STSL-CAT localizes the intended edits without manual intervention, which is necessary for the commercial software.

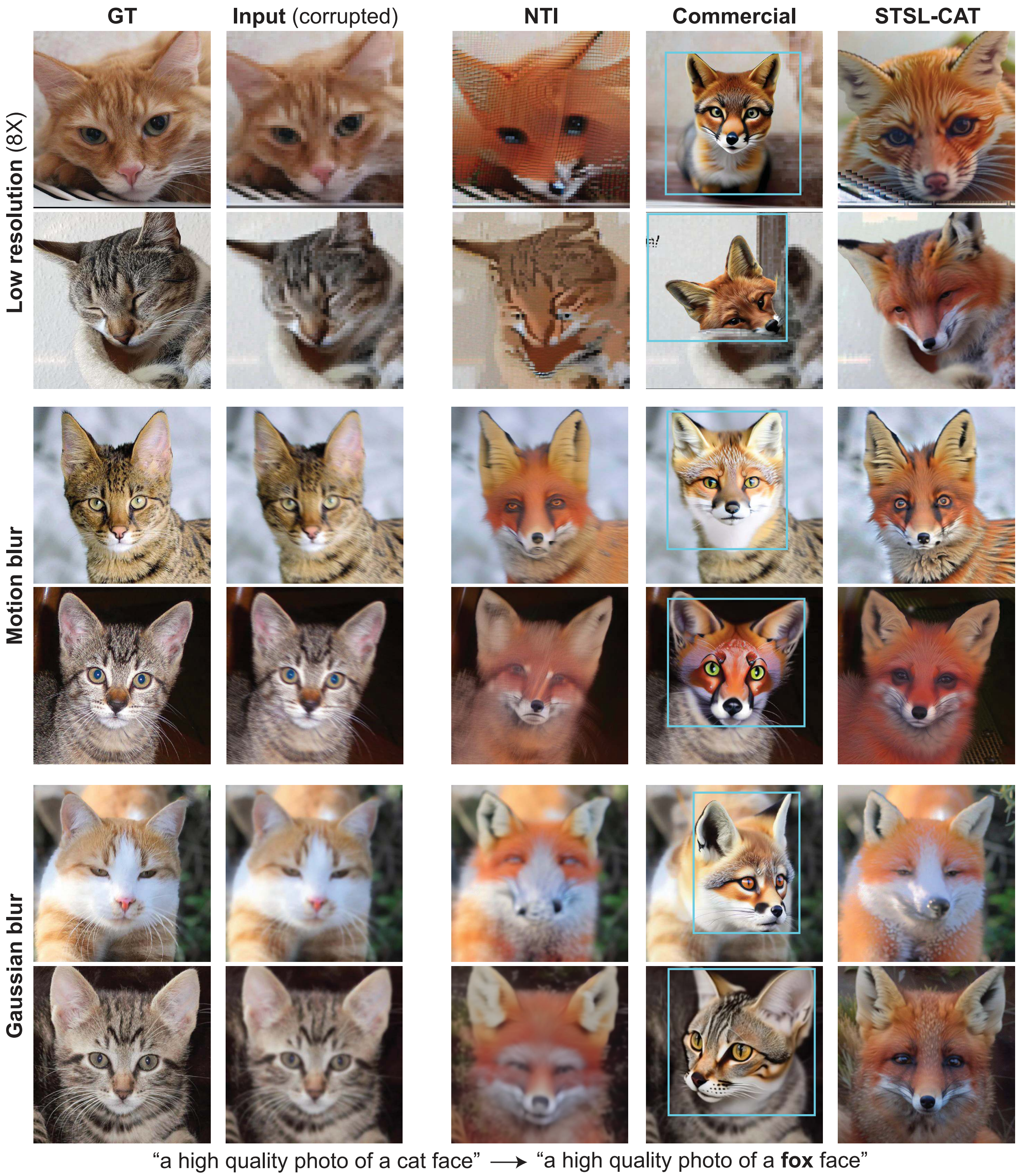

Figure 10. **Qualitative results on image editing on the corrupted images "cat" to "fox".** The proposed method STSL preserves the *content* of the source image while performing *text-guided image editing* on corrupt images.

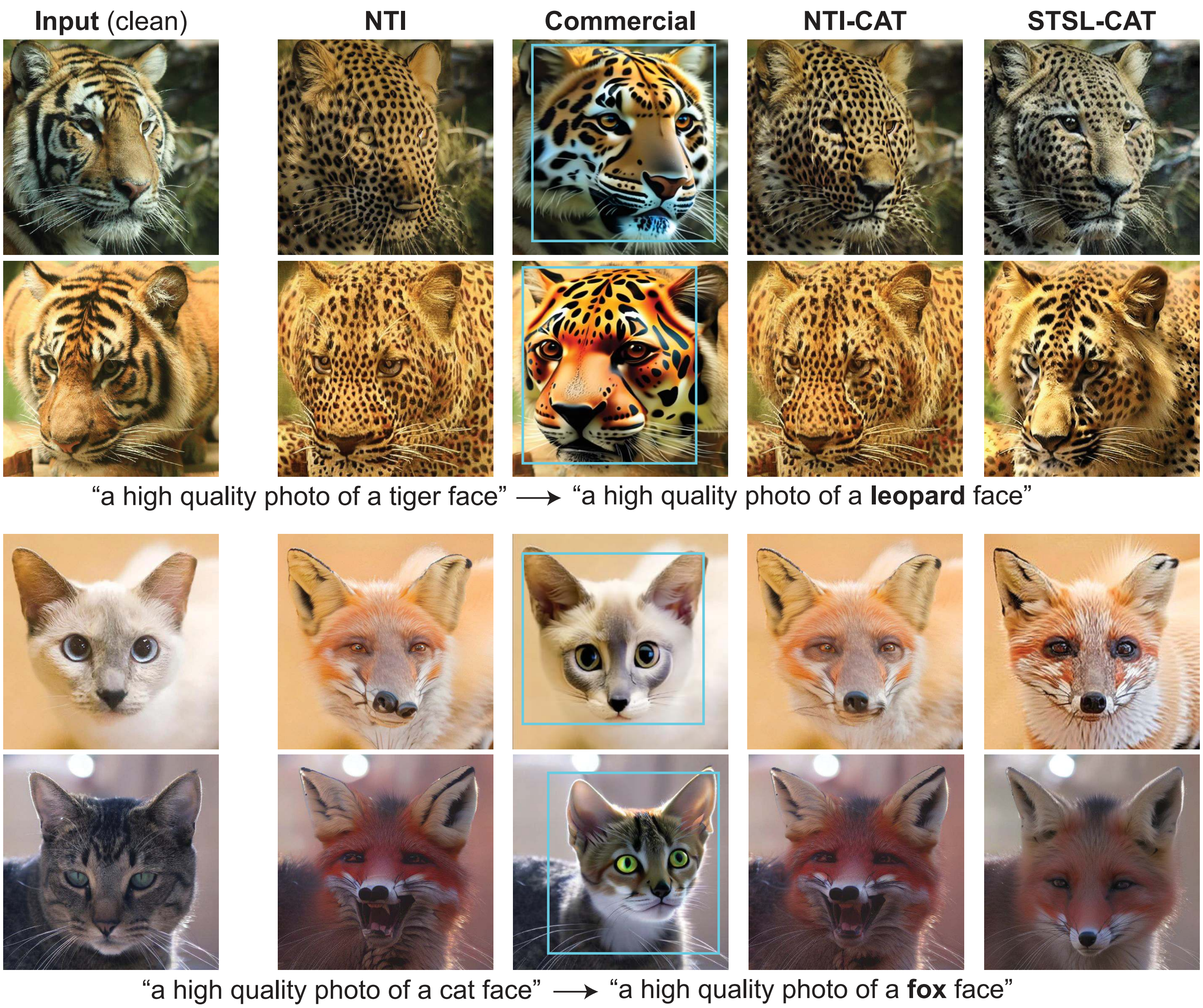

Figure 11. **Qualitative results on Image editing on the clean images.** Cross attention tuning (CAT) helps preserve image details with NTI [35] (NTI-CAT), and STSL-CAT further enhances the quality of the image by refining the forward latents.