# Prompt-tuning Latent Diffusion Models for Inverse Problems

**Hyungjin Chung** [1] **Jong Chul Ye** [1] **Peyman Milanfar** [2] **Mauricio Delbracio** [2]

## Abstract

We propose a new method for solving imaging inverse problems using text-to-image latent diffusion models as general priors. Existing methods using latent diffusion models for inverse problems typically rely on simple null text prompts, which can lead to suboptimal performance. To improve upon this, we introduce a method for prompt tuning, which jointly optimizes the text embedding on-the-fly while running the reverse diffusion. This allows us to generate images that are more faithful to the diffusion prior. Specifically, our approach involves a unified optimization framework that simultaneously considers the prompt, latent, and pixel values through alternating minimization. This significantly diminishes image artifacts - a major problem when using latent diffusion models instead of pixel-based diffusion ones. Our method, called P2L, outperforms both pixel- and latent-diffusion model-based inverse problem solvers on a variety of tasks, such as super-resolution, deblurring, and inpainting. Furthermore, P2L demonstrates remarkable scalability to higher resolutions without artifacts.

## 1. Introduction

Imaging inverse problems are often solved by optimizing or sampling a functional that combines a data-fidelity/likelihood term with a regularization term or signal prior (Romano et al., 2017; Venkatakrishnan et al., 2013; Ongie et al., 2020; Kamilov et al., 2023; Kawar et al., 2022; Kadkhodaie & Simoncelli, 2021; Chung et al., 2023b). A common regularization strategy is to use pre-trained image priors from generative models, such as GANs (Bora et al., 2017), VAEs (Bora et al., 2017; González et al., 2022), Normalizing flows (Whang et al., 2021) or Diffusion models (DM) (Song et al., 2022; Chung & Ye, 2022).

In particular, DMs have gained significant attention as implicit generative priors for solving inverse problems in imaging (Kadkhodaie & Simoncelli, 2021; Whang et al., 2022; Daras et al., 2022; Kawar et al., 2022; Feng et al., 2023; Laroche et al., 2023; Chung et al., 2023b). Leaving the pre-trained diffusion prior intact, one can guide the inference process to perform posterior sampling conditioned on the measurement at inference time by resorting to Bayesian inference. In the end, the ultimate goal of Diffusion model-based Inverse problem Solvers (DIS) would be to act as a fully general inverse problem solver, which can be used not only regardless of the imaging model, but also regardless of the data distribution.

Solving inverse problems in a fully general domain is hard. This directly stems from the difficulty of generative modeling a wide distribution, where it is known that one has to trade-off diversity with fidelity by some means of sharpening the distribution (Brock et al., 2018; Dhariwal & Nichol, 2021). The standard approach in modern DMs is to condition on text prompts (Rombach et al., 2022; Saharia et al., 2022b), among them the most popular being Stable Diffusion (SD), a latent diffusion model (LDM), which is itself an under-explored topic in the context of inverse problem solving. While text conditioning is now considered standard practice in content creation including images (Ramesh et al., 2022; Saharia et al., 2022b), 3D (Poole et al., 2023; Wang et al., 2023c), video (Ho et al., 2022), personalization (Gal et al., 2022), and editing (Hertz et al., 2022), it has been completely disregarded in the inverse problem solving context. This is natural, as it is highly ambiguous which text would be beneficial to use when all we have is a degraded measurement. The wrong prompt could easily lead to degraded performance.

In this work, we aim to bridge this gap by proposing a way to *automatically* find the right prompt to condition DMs when solving inverse problems. This can be achieved through optimizing the continuous text embedding *on-the-fly* while running DIS. We formulate this into a united framework of updating the text embedding and the latent in an alternating fashion, such that they become gradually aligned during the sampling process. Orthogonal and complementary to embedding optimization, we devise a simple Latent DIS

[1]KAIST, Daejeon, Korea [2]Google Research, Mountain View, US. Correspondence to: Hyungjin Chung <hj.chung@kaist.ac.kr>, Mauricio Delbracio <mdelbra@google.com>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

(LDIS) that controls the evolution of the latents to stay on the natural data manifold and additionally utilizes the VAE prior for stability of the solutions. We name the algorithm that combines these components P2L, short for **P**rompt-tuning **P**rojected **L**atent diffusion model-based inverse problem solver. In reaching for the ultimate goal of DIS, we focus on 1) **Latent DIS** (LDIS) for solving inverse problems in the 2) **fully general domain** (using a single pre-trained checkpoint) that targets 3) **512×512 resolution**[1]. All the aforementioned components are highly challenging, and thus not extensively studied.

## 2. Background

### 2.1. Latent diffusion models

DMs are generative models that learn to reverse the forward noising process (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021), starting from the initial distribution $p_0(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^n$ and approaching the standard Gaussian $p_T(\boldsymbol{x}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as $T \to \infty$ by the forward Gaussian perturbation kernels $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0, t^2\boldsymbol{I})$[2]. The forward/reverse processes can be characterized with Ito stochastic differential equations (SDE). Sampling from the distribution can either be done through solving the reverse SDE, or equivalently by solving the probability-flow ordinary differential equation (PF-ODE) (Song et al., 2021; Karras et al., 2022):

$$d\boldsymbol{x}_t = -t\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)\, dt = \frac{\boldsymbol{x}_t - \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]}{t}\, dt, \quad (1)$$

with $\boldsymbol{x}_T \sim p_T(\boldsymbol{x}_T)$, where we use the Tweedie's formula (Efron, 2011) given as $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \boldsymbol{x}_t + t^2\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$. Here $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ is typically approximated with a score network $\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot)$ or a noise estimation network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$, and learned through denoising score matching (DSM) (Vincent, 2011) or epsilon-matching loss (Ho et al., 2020).

Pixel DMs that operate on the pixel space $\boldsymbol{x}$ are compute-heavy. One workaround for compute-efficient generative modeling is to leverage a variational autoencoder that maximizes the evidence lower bound (ELBO) (Rombach et al., 2022; Kingma & Welling, 2013). This leads to the following encoder and decoder representation for all $\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}) \in \mathbb{R}^n$: $\boldsymbol{x} = \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z})$, where

$$\boldsymbol{z} = \mathcal{E}_{\boldsymbol{\phi}}(\boldsymbol{x}) := \mathcal{E}^{\boldsymbol{\mu}}_{\boldsymbol{\phi}}(\boldsymbol{x}) + \mathcal{E}^{\boldsymbol{\sigma}}_{\boldsymbol{\phi}}(\boldsymbol{x}) \odot \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}). \quad (2)$$

Here, $\mathcal{E}^{\boldsymbol{\mu}}_{\boldsymbol{\phi}}, \mathcal{E}^{\boldsymbol{\sigma}}_{\boldsymbol{\phi}}$ are parts of the encoder that outputs the mean and the variance of the encoder distribution, $\mathcal{D}_{\boldsymbol{\varphi}}$ is the decoder, and $\boldsymbol{z} \in \mathbb{R}^k$ with $k < n$ corresponds to the *latent* representation. After encoding into the latent space (Rombach et al., 2022), one can train a DM in the low-dimensional latent space. LDMs are beneficial in that the computation is cheaper as it operates in a lower-dimensional space, consequently being more suitable for modeling higher dimensional data (e.g. large images of size $\geq 512^2$). The effectiveness of LDMs have democratized the use of DMs as the de facto standard of generative models especially for images under the name of Stable Diffusion (SD), which we focus on extensively in this work.

One notable difference of SD from standard pixel DMs (Dhariwal & Nichol, 2021) is the use of text conditioning $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot, \mathcal{C})$, where $\mathcal{C}$ is the continuous embedding vector usually obtained through the CLIP text embedder (Radford et al., 2021). As the model is trained with LAION-5B (Schuhmann et al., 2022), a large-scale dataset containing image-text pairs, SD can be conditioned during the inference time to generate images that are aligned with the given text prompt by directly using $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot, \mathcal{C})$, or by means of classifier-free guidance (CFG) (Ho & Salimans, 2021).

### 2.2. Solving inverse problem with (L)DMs

Given access to some measurement

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n},\ \boldsymbol{A} \in \mathbb{R}^{m \times n},\ \boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 \boldsymbol{I}_m) \quad (3)$$

where $\boldsymbol{A}$ is the forward operator and $\boldsymbol{n}$ is additive white Gaussian noise, the task is retrieving $\boldsymbol{x} \in \mathbb{R}^n$ from $\boldsymbol{y} \in \mathbb{R}^m$. As the problem is ill-posed, a natural way to solve it is to perform posterior sampling $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{y})$ by defining a suitable prior $p(\boldsymbol{x})$. In DIS, DMs (i.e. denoisers) act as the implicit prior with the use of the score function. The objective of solving inverse problems is to provide a restoration that is as close as possible to the ground truth given the measurement, whether we are targeting to minimize the distortion or to maximize the perceptual quality (Blau & Michaeli, 2018; Delbracio & Milanfar, 2023).

Earlier methods utilized an alternating projection approach, where hard measurement constraints are applied in-between the denoising steps whether in pixel space (Kadkhodaie & Simoncelli, 2021; Song et al., 2021) or measurement space (Song et al., 2022; Chung & Ye, 2022). Distinctively, projection in the spectral space via singular value decomposition (SVD) to incorporate measurement noise has been developed (Kawar et al., 2021; 2022). Subsequently, methods that aim to approximate the gradient of the log posterior

[1] All prior works on DIS/LDIS focused on 256×256 resolution. Most LDIS focused their evaluation on a constrained dataset such as FFHQ, and did not scale their method to more general domains such as ImageNet.

[2] Here, we use the choice used in (Karras et al., 2022) for simplicity, but use variance preserving (VP) models (Song et al., 2021) for experiments as pre-trained models are available in this form. The different choices can be considered equivalent (Kawar et al., 2022)

in the diffusion model context have been proposed (Chung et al., 2023b; Song et al., 2023b), expanding the applicability to nonlinear problems. Broadening the range even further, methods that aim to solve blind (Chung et al., 2023a; Murata et al., 2023), 3D (Chung et al., 2023c; Lee et al., 2023), and unlimited resolution problems (Wang et al., 2023b) were introduced. More recently, methods leveraging diffusion score functions within variational inference to solve inverse imaging has been proposed (Mardani et al., 2023; Feng et al., 2023). Notably, all the aforementioned methods utilize *pixel-domain* DMs. Orthogonal to this direction, some of the recent works have shifted their attention to using *latent* diffusion models (Rout et al., 2023b; Song et al., 2023a; He et al., 2023), a direction that we follow in this work.

In fact, inverse solvers can be directly linked to posterior sampling from $p(\boldsymbol{x}_0|\boldsymbol{y})$, which can be achieved by modifying Eq. (1) with

$$d\boldsymbol{x}_t = -t\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y})\, dt = \frac{\boldsymbol{x}_t - \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]}{t}\, dt. \tag{4}$$

Here, $\log p(\boldsymbol{x}_t|\boldsymbol{y}) = \log p(\boldsymbol{x}_t) + \log p(\boldsymbol{y}|\boldsymbol{x}_t)$. However, as $\log p(\boldsymbol{y}|\boldsymbol{x}_t)$ is intractable, DPS (Chung et al., 2023b) proposes to approximate it with $\log p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq \log p(\boldsymbol{y}|\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t])$, whose approximation error can be quantified and bounded by the Jensen gap.

This idea was recently extended to LDMs in a few recent works (Rout et al., 2023b; He et al., 2023), which consider the following straightforward extension image domain DPS (Chung et al., 2023b) as the baselne.

$$\begin{aligned}\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\boldsymbol{z}_t) &\simeq \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\mathcal{D}_{\boldsymbol{\varphi}}(\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t])) \\ &= \nabla_{\boldsymbol{z}_t} \|\boldsymbol{y} - \mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_0)\|_2^2/\sigma_y^2, \end{aligned} \tag{5}$$

with $\hat{\boldsymbol{z}}_0 := \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t]$, leading the following latent update:

$$\boldsymbol{z}_{t-1} = \mathrm{DDIM}(\boldsymbol{z}_t) - \rho\nabla_{\boldsymbol{z}_t} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\varphi}(\hat{\boldsymbol{z}}_0)\|_2, \tag{6}$$

where $\rho$ is the step size, and $\mathrm{DDIM}(\cdot)$ denotes a single step of DDIM (or DDPM in general) sampling. We refer to the sampler that uses the approximation in Eq. (5) as Latent DPS (LDPS) henceforth.

However, the crucial component that delineates LDM is the existence of VAE. When naively using the LDPS in Eq. (6), the decoder introduces a significant amount of error especially when the estimated clean latent $\hat{\boldsymbol{z}}_0^{(\mathcal{C})}$ falls off the manifold of the clean latents. To address this, (Rout et al., 2023b) proposed Posterior Sampling using Latent Diffusion (PSLD) to regularize the update steps on the latent so that the clean latents are led to the fixed point of the successive application of decoding-encoding. Formally, omitting the dependence on $\mathcal{C}$, they use the following gradient step

$$\begin{aligned}\nabla_{\boldsymbol{z}_t} \log(\boldsymbol{y}|\boldsymbol{z}_t) \simeq \nabla_{\boldsymbol{z}_t}(&\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_0)\|_2^2 + \\ &\lambda\|\hat{\boldsymbol{z}}_0 - \mathcal{E}_{\boldsymbol{\phi}}(\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_0))\|_2^2), \end{aligned} \tag{7}$$

where the additional regularization term weighted by $\lambda$ leads $\hat{\boldsymbol{z}}_0$ towards the fixed point. On the other hand, (He et al., 2023) extends LDPS by using history updates as in Adam (Kingma & Ba, 2015). Concurrent work by Rout et al. (2023a) proposes to match higher-order moments of Tweedie.

The adoption of a regularized version, as indicated in Eq. (7), over the baseline formulation Eq. (5) presents a compromise between maintaining data fidelity and ensuring the stability of the VAE. This often leads to a decline in performance, particularly in scenarios with low SNR, as will be demonstrated in subsequent experimental results[3]. Furthermore, most existing works in the literature that aim for LDIS, to the best of our knowledge, neglect the use of text embedding by resorting to the use of null text embedding $\mathcal{C}_{\varnothing}$. There exists one concurrent work (Kim et al., 2023) which uses a *fixed* target text while adapting the null text in CFG to emphasize the target text conditioning when solving inverse problems. Our method is orthogonal to Kim et al. (2023) as our aim is to automatically find the ambiguous text embedding that best describes the image, rather than guide the result towards a specific mode described by the target text.

### 2.3. Prompt-tuning inverse problem solver

In modern language models and vision-language models, *prompting* is a standard technique (Radford et al., 2021; Brown et al., 2020) to guide the large pre-trained models to solve downstream tasks. As it has been found that even slight variations in the prompting technique can lead to vastly different outcomes (Kojima et al., 2022), prompt tuning (learning) has been introduced (Shin et al., 2020; Zhou et al., 2022), which defines a *learnable* context vector to optimize over. It was shown that by only optimizing over the continuous embedding vector while maintaining the model parameters fixed, one can achieve a significant performance gain. In the context of DMs, prompt tuning has been adopted for personalization (Gal et al., 2022; Mokady et al., 2023), where one defines a special token to embed a specific concept with only a few images.

Inspired by this, we are interested in the prompt optimization in LDIS. In the context of LDIS,

$$\arg\min_{\boldsymbol{x},\boldsymbol{c}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{c}) \equiv \arg\min_{\boldsymbol{z},\boldsymbol{c}} \mathcal{L}(\mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z}), \boldsymbol{c}) \tag{8}$$

[3] Also see Fig. 13 and Fig. 14, where it can be seen that repeatedly applying encoding-decoding steps yields diverging results, regardless of using "glue" steps introduced in PSLD (Rout et al., 2023b).

| | FFHQ | | | | | | ImageNet | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SR×8 | | | Inpaint ($p = 0.8$) | | | SR×8 | | | Inpaint ($p = 0.8$) | | |
| **Prompt** | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ |
| `""` | 61.16 | 0.327 | 26.49 | 52.34 | 0.241 | **29.78** | 78.68 | 0.397 | 23.49 | 70.87 | 0.350 | 26.20 |
| `"A high quality photo"` | 61.17 | 0.327 | 26.57 | 52.82 | 0.237 | 29.70 | 77.00 | 0.396 | 23.51 | 69.10 | 0.350 | 26.26 |
| `"A high quality photo of a cat"` | 69.03 | 0.377 | 26.39 | 55.15 | 0.248 | 29.63 | 76.69 | 0.402 | **23.63** | 68.48 | 0.355 | 26.13 |
| `"A high quality photo of a dog"` | 66.55 | 0.371 | 26.48 | 55.91 | 0.249 | 29.65 | 76.45 | 0.394 | 23.58 | 67.75 | 0.354 | 26.10 |
| `"A high quality photo of a face"` | 60.41 | 0.325 | 26.74 | 52.33 | 0.239 | 29.69 | 77.32 | 0.403 | 23.60 | 68.83 | 0.352 | 26.20 |
| Prompt optimizaton | **58.73** | **0.317** | 26.68 | **51.40** | **0.233** | 29.69 | 66.96 | **0.386** | 23.57 | 66.82 | **0.314** | **26.29** |
| PALI prompts from $\boldsymbol{y}$ | 61.33 | 0.329 | **26.81** | 54.34 | 0.249 | 29.76 | 68.28 | 0.388 | 23.57 | 69.55 | 0.355 | 26.26 |
| PALI prompts from $\boldsymbol{x}$ | 60.73 | 0.322 | 26.76 | 52.06 | 0.238 | 29.75 | **66.55** | 0.387 | 23.57 | **64.00** | 0.348 | 26.17 |

*Table 1.* Difference in restoration performance using LDPS on SR×8 task with varying text prompts. Prompt optimization: text embedding optimized without access to ground truth. PALI prompts from $\boldsymbol{x}$/$\boldsymbol{y}$: captions are generated with PALI (Chen et al., 2022) from $\boldsymbol{x}$: ground truth clean images / $\boldsymbol{y}$: degraded images. The former can be considered an empirical upper bound.

where the first equation follows from $\boldsymbol{x} = \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z})$ in the deterministic decoder mapping of VAE, where $\boldsymbol{c}$ is the text embedding and the loss $\mathcal{L}$ will be explained in more detail in subsequent session. It is easy to see that

$$\arg\min_{\boldsymbol{z},\boldsymbol{c}} \mathcal{L}(\mathcal{D}_{\varphi}(\boldsymbol{z}), \boldsymbol{c}) \leq \arg\min_{\boldsymbol{z}} \mathcal{L}(\mathcal{D}_{\varphi}(\boldsymbol{z}), \boldsymbol{c} = \mathcal{C}_{\varnothing}), \quad (9)$$

where $\mathcal{C}_{\varnothing}$ is the text embedding from the null text prompt. Notably, by keeping one of the variables fixed, we are optimizing for the *upper bound* of the objective that we truly wish to optimize over. It would be naturally beneficial to optimize the LHS of Eq. (9), rather than the RHS used in the previous methods.

To see Eq. (9) in effect, we conduct two canonical experiments with 256 test images of FFHQ (Karras et al., 2019) and ImageNet (Deng et al., 2009): super-resolution (SR) of scale $\times 8$ and inpainting with 80% of the pixels randomly dropped, using the LDPS algorithm. Keeping all the other hyper-parameters fixed, we only vary the text condition for the diffusion model. In addition to using a general text prompt, we use PALI (Chen et al., 2022) to provide captions from the ground truth images ($\boldsymbol{x}$) and from the measurements ($\boldsymbol{y}$) and use them when running LDPS. Further experimental details can be found in Appendix B. In Table 1, we first see that simply varying the text prompts can lead to dramatic difference in the performance. For instance, we see an increase of over **10 FID** when we use the text prompts from PALI for the task of $\times 8$ SR on ImageNet. In contrast, using the prompts generated from $\boldsymbol{y}$ often degrades the performance (e.g. inpainting) as the correct captions cannot be generated. Indeed, from the table, we see that by applying our prompt tuning approach, we achieve a large performance gain, sometimes even outperforming the PALI captions which has full access to the ground truth when attaining the text embeddings. From this motivating example, it is evident that additionally optimizing for $\boldsymbol{c}$ would bring us gains that are orthogonal to the development of the solvers (Rout et al., 2023b; He et al., 2023; Song et al., 2023a), a direction which will be explored in this paper.

## 3. Main Contribution: the P2L algorithm

To effectively utilize the Latent Diffusion Inverse Solver (LDIS) with prompt optimization, it is crucial to ensure two key criteria: 1) consistency with respect to the measurements, and 2) the feasibility of the latent as per the LDM. Our approach diverges from conventional regularization strategies, such as PSLD in Eq. (7). Instead, we base our formulation on Eq. (9), which offers a more direct route to achieving these objectives:

$$\min_{\boldsymbol{z}\in P(\boldsymbol{z}|\boldsymbol{y})} \min_{\mathcal{C}} \quad \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\varphi}(\boldsymbol{z}^{(\mathcal{C})})\|^2 \quad (10)$$

$$\text{subject to} \quad \boldsymbol{z} \in F_X \quad (11)$$

where $P(\boldsymbol{z}|\boldsymbol{y})$ denotes the posterior distribution of $\boldsymbol{z}$ given the measurement condition $\boldsymbol{y}$ and $F_X$ denotes the set of latent that can be represented by some image $x$:

$$F_X = \{\boldsymbol{z} | \boldsymbol{z} = \mathcal{E}_{\phi}^{\mu}(\boldsymbol{x}) \text{ for some } \boldsymbol{x}\}$$

A key contribution of our study is the demonstration that the optimization problem involving prompt, latent, and pixel values can be effectively addressed through alternating minimization, as explained in the following sections. We summarize our alternating sampling method in Algorithm 1 and Algorithm 2, based on DDIM sampling, with standard noise schedule notations adopted from (Ho et al., 2020).

The intuition of the overall algorithm is that by incorporating the text conditioning automatically, ambiguities arising from the natural ill-posedness of the inverse problems can be mitigated. Further, artifacts that often arise from naive latent space optimization can be corrected by leveraging the VAE during inverse problem-solving. Brief overview of the method structure:

1. (Sec. 3.1) Prompt embedding optimization through fidelity loss minimization

**Algorithm 1** P2L

**Require:** $\boldsymbol{\epsilon_\theta}, \boldsymbol{z}_T, \boldsymbol{y}, \mathcal{C}, T, K, \gamma, \lambda_{\mathcal{D}}$
1: **for** $t = T$ **to** 1 **do**
2: $\quad \mathcal{C}_t^* \leftarrow \textsc{OptimizeEmb}(\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}_t^0, K)$ ▷ Sec. 3.1
3: $\quad \hat{\boldsymbol{\epsilon}}_t \leftarrow \boldsymbol{\epsilon_\theta}(\boldsymbol{z}_t, \mathcal{C}_t^*)$
4: $\quad \hat{\boldsymbol{z}}_{0|t} \leftarrow (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)/\sqrt{\bar{\alpha}_t}$
5: $\quad$ **if** $(t \mod \gamma) = 0$ **then**
6: $\quad\quad \hat{\boldsymbol{x}}_0 \leftarrow \arg\min_{\boldsymbol{x}_0} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_0\|_2^2 + \lambda\|\boldsymbol{x}_0 - \mathcal{D}_\varphi(\hat{\boldsymbol{z}}_{0|t})\|_2^2$
7: $\quad\quad \tilde{\boldsymbol{z}}_{0|t} \leftarrow \mathcal{E}_{\boldsymbol{\phi}}(\hat{\boldsymbol{x}}_0)$ ▷ Sec. 3.3
8: $\quad$ **else**
9: $\quad\quad \tilde{\boldsymbol{z}}_{0|t} \leftarrow \hat{\boldsymbol{z}}_{0|t}$
10: $\quad$ **end if**
11: $\quad \boldsymbol{z}'_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\tilde{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_t$
12: $\quad \boldsymbol{z}_{t-1} \leftarrow \boldsymbol{z}'_{t-1} - \rho_t \nabla_{\boldsymbol{z}_t}\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_{0|t})\|$ ▷ Sec. 3.2
13: $\quad \mathcal{C}_{t-1}^{(0)} \leftarrow \mathcal{C}_t^*$
14: **end for**
15: **return** $\boldsymbol{x}_0 \leftarrow \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z}_0)$

**Algorithm 2** Prompt tuning

1: **function** $\textsc{OptimizeEmb}(\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}_t^{(0)}, K)$
2: $\quad$ **for** $k = 1$ **to** $K$ **do**
3: $\quad\quad \hat{\boldsymbol{\epsilon}}_t \leftarrow \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\boldsymbol{z}_t, \mathcal{C}_t^{(k-1)})$
4: $\quad\quad \hat{\boldsymbol{z}}_{0|t} \leftarrow (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)/\sqrt{\bar{\alpha}_t}$
5: $\quad\quad \hat{\boldsymbol{z}}'_{0|t} \leftarrow \hat{\boldsymbol{z}}_{0|t} - \rho\nabla_{\hat{\boldsymbol{z}}_{0|t}}\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_{0|t})\|$
6: $\quad\quad \mathcal{L}_t \leftarrow \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}'_{0|t}, \mathcal{C}_t^{(k-1)})\|_2^2$
7: $\quad\quad \mathcal{C}_t^{(k)} \leftarrow \mathcal{C}_t^{(k-1)} - \texttt{AdamGrad}(\mathcal{L}_t)$
8: $\quad$ **end for**
9: $\quad$ **return** $\mathcal{C}_t^* \leftarrow \mathcal{C}_t^{(K)}$
10: **end function**

2. (Sec. 3.2) Latent update via LDPS step with the optimized prompt

3. (Sec. 3.3) Latent correction. This involves decoding, enforcing data consistency in the pixel space, and re-encoding

### 3.1. Prompt tuning

To update the prompt, we address the inner optimization challenge presented in Eq. (10), which involves identifying a suitable latent that aligns with $z \in P(\boldsymbol{z}|\boldsymbol{y})$. This process is akin to the approach used in decomposed diffusion sampling (DDS) as described in (Chung et al., 2024). It entails minimizing the loss detailed in Eq. (10), starting from the denoised latent $\hat{\boldsymbol{z}}_{0|t}$. As a result, we obtain $\hat{\boldsymbol{z}}'_{0|t}$ as a first order approximation of $\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}]$, by adjusting $\hat{\boldsymbol{z}}_{0|t}$ and incorporating data consistency, as outlined in Line 5 of Algorithm 2. Now for the updated latent $\hat{\boldsymbol{z}}'_{0|t}$,we should solve the inner optimization problem of Eq. (10), leading to the following optimization:

$$\mathcal{C}_t^* = \arg\min_{\mathcal{C}} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_{0|t}^{(\mathcal{C})})\|_2^2 \tag{12}$$

We found that this alternating minimization should be solved multiple times to have meaningful update of the problem. This corresponds to the OPTIMIZEEMB in Algorithm 1, with details of the optimization function in Algorithm 2. Further details can be found in Appendix A,D.

### 3.2. Enforcing data fidelity

For a given optimized prompt $\mathcal{C}_t^*$, a straightforward extension of the LDPS for latent update $\boldsymbol{z}_{t-1}$ in Eq. (6) is

$$\boldsymbol{z}_{t-1} = \text{DDIM}(\boldsymbol{z}_t) - \rho\nabla_{\boldsymbol{z}_t}\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_\varphi(\hat{\boldsymbol{z}}_0^{(\mathcal{C}_t^*)})\|_2^2, \tag{13}$$

where $\hat{\boldsymbol{z}}_0^{(\mathcal{C}_t)} := \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t, \mathcal{C}_t^*]$ is the prompt conditioned posterior mean. Here, $\text{DDIM}(\boldsymbol{z}_t)$ can be equivalently represented by the denosing step through Tweedie's formula:

$$\hat{\boldsymbol{z}}_{0|t} := (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)/\sqrt{\bar{\alpha}_t} \tag{14}$$

followed by the noising step:

$$\text{DDIM}(\boldsymbol{z}_t) = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_t \tag{15}$$

where $\hat{\boldsymbol{\epsilon}}_t := \boldsymbol{\epsilon_\theta}(\boldsymbol{z}_t, \mathcal{C}_t^*)$, as shown in (Chung et al., 2024). These data fidelity enforcing steps are presented in line 3-4,9-12 of Algorithm 1.

### 3.3. Enforcing latent feasibility

However, the aforementioned data fidelity enforcing steps do not consider the latent constraint Eq. (11). Specifically, without considering our constraint, we see in Fig. 2 and Fig. 3 that artifacts arise, and this cannot be fully mitigated by leveraging the regularizations proposed in PSLD (Rout et al., 2023b). In this section, we show that this constraint can be easily enforced by incorporating the VAE prior.

Specifically, inspired by the regularization term in PSLD in Eq. (7), we consider the following loss, which is the maximum a posteriori (MAP) objective under the VAE prior in Eq. (2) with isotropic covariance (González et al., 2022).

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{z}) = \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_\varphi(\boldsymbol{z})\|_2^2 + \zeta\|\boldsymbol{z} - \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\boldsymbol{x})\|_2^2, \tag{16}$$

where $\zeta$ absorbs the weighting caused by the variance of respective terms. Instead of enforcing the *hard* constraint between the $\boldsymbol{x}$ and $\boldsymbol{z}$ in the form of $\boldsymbol{x} = \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z})$ as in (Rout et al., 2023b) that can introduce the trade-off between data consistency and the stability of latent, our main goal is to enforce a *soft* constraint by splitting the variables similar in spirit to the alternating direction method of multipliers (ADMM) (Boyd et al., 2011).

Namely, using the variable splitting $\boldsymbol{x} = \mathcal{D}_\varphi(\boldsymbol{z})$, the optimization problem with respect to $\boldsymbol{x}$ becomes

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \zeta\|\boldsymbol{z} - \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\mathcal{D}_\varphi(\boldsymbol{z}))\|_2^2 + \lambda\|\boldsymbol{x} - \mathcal{D}_\varphi(\boldsymbol{z}) + \boldsymbol{\eta}\|_2^2. \quad (17)$$

where $\boldsymbol{\eta}$ denotes the dual variable in ADMM. Since we consider using only a single step ADMM update for each diffusion sampling step, we set dual variable $\boldsymbol{\eta}$ as a zero vector and do not consider its update. This leads to

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x} - \mathcal{D}_\varphi(\boldsymbol{z})\|_2^2. \quad (18)$$

Solving for Eq. (18) is performed using conjugate gradient (CG) with the *clean* latent $\hat{\boldsymbol{z}}_{0|t}$ obtained through the Tweedie's formula Eq. (14), leading to the following update:

$$\hat{\boldsymbol{x}}_0 = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x} - \mathcal{D}_\varphi(\hat{\boldsymbol{z}}_{0|t})\|_2^2 \quad (19)$$

as presented in line 3-6 of Algorithm 1. The resulting optimization problem in Eq. (19) is indeed a pixel-domain proximal update from $\mathcal{D}_\varphi(\hat{\boldsymbol{z}}_{0|t})$, which can be interpreted as enforcing the data fidelity in the pixel domain under the regularization from latent feasiblity.

Subsequently, using the encoder approximation and setting $\boldsymbol{z} = \mathcal{E}_{\boldsymbol{\phi}}(\boldsymbol{x})$ with $\boldsymbol{\eta} = \boldsymbol{0}$, the optimization problem with respect to $\boldsymbol{z}$ reads

$$\min_{\boldsymbol{z}} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}\mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\boldsymbol{x})\|_2^2 + \zeta\|\boldsymbol{z} - \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\boldsymbol{x})\|_2^2.$$

For a given pixel-domain update $\hat{\boldsymbol{x}}_0$ from Eq. (19), the corresponding latent update then has the closed-form solution

$$\hat{\boldsymbol{z}}_{0|t} = \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\hat{\boldsymbol{x}}_0) \quad (20)$$

Note that by Eq. (20), we guarantee that the clean latents stay on the *range space* of the encoder, automatically satisfying the constraint in Eq. (11). For this reason, we often denote the method proposed in this section simply as "projection" to the constraint set.

In practice, we choose to apply Eq. (19) and Eq. (20) every few iteration to control dramatic changes in the sampling, and to save computation. Nevertheless, solving Eq. (19) requires access to $\boldsymbol{A}^\top$, which is often non-trivial to define. Contrarily, our `jax` implementation enables defining $\boldsymbol{A}^\top$ through `jax.vjp`. For further discussion, see Appendix E. Upon implementing Eq. (19) and Eq. (20), we reintroduce a data consistency step, as demonstrated in lines 11-12 of Algorithm 1. This step is to ensure that the process does not deviate from data consistency.

### 3.4. Targetting arbitrary resolution

Another important contribution of this work is its scalability to arbitrary resolution with large image size. Despite its fully convolutional nature, as SD was trained with 64×64 latents ($\leftrightarrow$ 512 × 512 images), the performance degrades when we aim to deal with larger dimensions, again due to train-test time discrepancy. Several works aimed to mitigate this issue by processing the latents with strided patches (Bar-Tal et al., 2023; Jiménez, 2023; Wang et al., 2023a) that increases the computational burden by roughly $\mathcal{O}(n^2)$. In contrast, in Appendix F, we show that our approach using the projection step by simply running Alg. 1, used *without* any patch processing, can outperform previous methods that rely on patches, resulting in significantly improved image quality and faster inference speed. This is because when given a latent that stays within the range space of the encoder thanks to Eq. (20), the decoder is able to produce a high-quality image directly even when the input size is larger than 64× 64.

**Guidance on hyperparameter selection** P2L, with prompt embedding as an additional variable to optimize over, has more hyperparameters than standard DIS. While we report the best choices in Tab. 6, here we provide a solid choice that works well across most experiments. 1) For optimizing prompt embedding, 1 5 iterations ($K$) with a learning rate of $1e-4$ yields stable performance. We observe setting too high values of $K$ or learning rate leads to overfitting, while setting them too small yields marginal improvements. 2) One can reliably choose GD with static step size of 1.0 for LDPS update, as advised in many previous works (Chung et al., 2023b; Rout et al., 2023b). 3) projection works when applied every 3-5 steps ($\gamma$), while the value of $\lambda$ matters less and can be freely chosen between 0.1 1.0 with negligible difference in the performance. When applying projection too often, artifacts arise.

## 4. Experiments

**Datasets, Models** We consider two different well-established datasets: 1) FFHQ 512×512 (Karras et al., 2019), and 2) ImageNet 512×512 (Deng et al., 2009). For the former, we use the first 1000 images for testing, similar to (Chung et al., 2023b). For the latter, we choose 1k images out of 10k test images provided in (Saharia et al., 2022a) by interleaved sampling, i.e. using images of index 0, 10, 20, etc. after ordering by name. For the latent diffusion model, we choose SD v1.4 pre-trained on the LAION dataset for all the experiments, including the baseline comparison methods based on LDM. As there is no publicly available image diffusion model that is trained on an identical dataset, we choose ADM (Dhariwal & Nichol, 2021) trained on ImageNet 512×512 data as the universal prior

when implementing baseline pixel-domain DIS. Note that this discrepancy may lead to an unfair advantage in the performance for evaulation on ImageNet, and an unfair disadvantage in the performance when evaluating on FFHQ. All experiments were done on NVIDIA A100 40GB GPUs.

**Inverse Problems** We test our method on the following degradations: 1) Super-resolution from $\times 8$ averagepooling, 2) Inpainting from 10-20% free-form masking as used in (Saharia et al., 2022a), 3) Gaussian deblurring from an image convolved with a $61\times61$ size Gaussian kernel with $\sigma = 3.0$, 4) Motion deblurring from an image convolved with a $61\times61$ motion kernel that is randomly sampled with intensity 0.5[4], following (Chung et al., 2023b). For all degradations, we include mild additive white Gaussian noise with $\sigma_y = 0.01$.

**Evaluation** As the main objective of this study is to improve the performance of LDIS, we mainly focus our evaluation on the comparison against the current SOTA LDIS: we compare against LDPS, GML-DPS (Rout et al., 2023b), PSLD (Rout et al., 2023b), and LDIR (He et al., 2023). We additionally compare against TRreg (Kim et al., 2023) to emphasize that the aim of the works are different. All LDIS including the proposed P2L use 1000 NFE DDIM sampling with $\eta = 0.0$[5], with the exception of TReg, which uses 200 NFE DDIM sampling. Using higher NFE did not help in improving sample quality. We additionally compare against SOTA pixel-domain DIS: DPS (Chung et al., 2023b), Diff-PIR (Zhu et al., 2023), DDS (Chung et al., 2024), and ΠGDM (Song et al., 2023b). For DPS, we use 1000 NFE DDIM sampling. For Diff-PIR, DDS, and ΠGDM, we use 100 NFE DDIM sampling. We choose the optimal $\eta$ values for these algorithms through grid-search. Details about the comparison methods can be found in Appendix D.3. We perform a quantitative evaluation with standard metrics: PSNR, FID, and LPIPS.

**Comparison against baseline** In all of the inverse problems that we consider in the paper, our method outperforms all the baselines by quite a large margin in terms of perceptual quality, measured by FID and LPIPS, while keeping the distortion at a comparable level against the current state-of-the-art methods. Especially, we see about 10 FID decrease in deblurring and inpainting tasks compared to the runner up in both FFHQ and ImageNet dataset (See Tables 8,2). The superiority can also be clearly seen in Fig. 1, where P2L achieves stable, high-quality reconstruction throughout all tasks. Results from both LDPS and PSLD often contain local grid-like artifacts (Red boxes in Figures) and are blurry. With P2L, the restored images are sharpened while the artifacts are effectively removed. LDIR are less prone to artifacts owing to the smoothed history gradient updates, but often results in unrealistic textures and deviations from the measurement, which is also reflected in having the lowest PSNR among the LDIS-class methods. In contrast, P2L is free from such drawbacks even when leveraging Adam-like gradient update steps. It should be noted that the compute time for P2L linearly increases as we increase the number of training iterations for the text embedding. The compute time for $K = 0$ is similar to other LDIS baselines, but it becomes slower if $K$ becomes larger. Devising a more time-efficient way to perform text embedding optimization is thus a promising future research direction. For further details on the runtime analysis, see Appendix C.

One rather surprising finding is the heavy downgrade in the performance for DIS methods. Even on in-distribution ImageNet test data, methods such as DPS and DiffPIR become very unstable. This can be attributed to the generative prior being poor: directly training DMs on high-resolution images often result in poor performance[6]. This observation again points to the importance of developing methods that can leverage foundation models when aiming for general domain higher-resolution data. See Appendix G for further results. As a final note, we believe that the compromise in PSNR is related to the imperfectness of the VAE used in SD v1.4[7], and we expect such degradation to be mitigated when switching to better, larger autoencoders such as SDXL (Podell et al., 2023).

**Design components** In Table 3, we perform an ablation study on the design components of the proposed method. From the table, we confirm that prompt tuning, projection to the range space of the encoder, and performing proximal update step (denoted as $\mathbf{\Gamma}$) before the projection all contributes to the gain in the performance. It is important that these gains are synergistic, and one component does not hamper the other. In the Appendix Tab. 7, we further show that our prompt-tuning approach is robust to the variation in the hyper-parameters (learning rate, number of iterations). Specifically, among the 9 configurations that we try, only the one with 5 iterations, lr=0.001 is inferior to not using prompt tuning. In Fig. 3, we visualize the progress of $\mathcal{D}(\hat{\boldsymbol{z}}_0)$ through time $t$ starting from the same random seed, comparing LDPS, PSLD, and LDPS + projection (row 4 of Tab. 6). Here, we see that our proposed projection approach effectively suppresses the artifacts that arise during the reconstruction process, whereas PSLD introduces additional artifacts. Furthermore, in Appendix F, we show that our

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

[5] The parameter $\eta$ indicates the stochasticity of the sampler. $\eta = 0.0$ leads to deterministic PF-ODE.

[6] For $\geq 512 \times 512$ resolution, either using latent diffusion or using cascaded models (Saharia et al., 2022b) are popular.

[7] Auto-encoding 1000 ground-truth test images result in the following metrics: FFHQ (PSNR): 29.66 $\pm$ 2.29, ImageNet (PSNR): 27.12 $\pm$ 4.38.


*Figure 1.* Inverse problem solving results on ImageNet $512 \times 512$ test set. Row 1: SR$\times$8, Row 2: gaussian deblurring, Row 3: motion deblurring, row 4: inpainting.

| | SR ($\times$8) | | | Deblur (motion) | | | Deblur (gauss) | | | Inpaint | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ |
| P2L (ours) | **51.81** | **0.386** | **23.38** | **54.11** | **0.360** | **24.79** | **39.10** | **0.325** | 25.11 | **32.82** | **0.229** | <u>21.99</u> |
| LDPS | 61.09 | 0.475 | <u>23.21</u> | 71.12 | 0.441 | 23.32 | 48.17 | 0.392 | 24.91 | 46.72 | 0.332 | 21.54 |
| GML-DPS (Rout et al., 2023b) | 60.36 | <u>0.456</u> | <u>23.21</u> | 59.08 | 0.403 | 24.35 | <u>45.33</u> | 0.377 | **25.44** | 47.30 | 0.294 | 21.12 |
| PSLD (Rout et al., 2023b) | 60.81 | 0.471 | 23.17 | 59.63 | 0.398 | 24.21 | 45.44 | <u>0.376</u> | <u>25.42</u> | <u>40.57</u> | <u>0.251</u> | 20.92 |
| LDIR (He et al., 2023) | 63.46 | 0.480 | 22.23 | 88.51 | 0.475 | 21.37 | 72.10 | 0.506 | 22.45 | 50.65 | 0.313 | **23.28** |
| TReg (Kim et al., 2023) | 104.3 | 0.520 | 18.97 | 102.97 | 0.501 | 19.06 | 117.3 | 0.455 | 16.84 | 77.76 | 0.349 | 14.98 |
| DDS (Chung et al., 2024) | 203.2 | 1.213 | 12.72 | 84.67 | 0.925 | 14.52 | 70.51 | 0.835 | 16.58 | 60.18 | 0.354 | 17.03 |
| DPS (Chung et al., 2023b) | 54.61 | 0.544 | 20.70 | 71.99 | 0.599 | 19.62 | 98.33 | 0.910 | 15.05 | 71.70 | 0.360 | 15.15 |
| DiffPIR (Zhu et al., 2023) | 488.3 | 1.182 | 13.44 | 87.04 | 0.622 | 19.32 | 79.31 | 0.755 | 20.55 | 45.97 | 0.300 | 20.11 |
| ΠGDM (Song et al., 2023b) | <u>53.00</u> | 0.490 | 21.08 | 75.35 | 0.682 | 18.66 | 70.26 | 0.797 | 21.96 | 65.75 | 0.322 | 16.84 |

*Table 2.* Quantitative evaluation (PSNR, LPIPS, FID) of inverse problem solving on ImageNet 512$\times$512-1k validation dataset. **Bold**: best, <u>underline</u>: second best. Methods that are not LDM-based are shaded in gray.

| Design components | | | FFHQ | | | | ImageNet | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | SR$\times$8 | | Inpaint ($p = 0.8$) | | SR$\times$8 | | Inpaint ($p = 0.8$) | |
| Projection | $\Gamma$ | Prompt tuning | FID↓ | PSNR↑ | FID↓ | PSNR↑ | FID↓ | PSNR↑ | FID↓ | PSNR↑ |
| ✗ | ✗ | ✗ | 61.16 | 26.49 | 52.34 | **29.78** | 78.68 | 23.49 | 70.87 | 26.20 |
| ✗ | ✗ | ✓ | 58.73 | **26.68** | 51.40 | 29.69 | 76.40 | **23.52** | 67.06 | 26.32 |
| ✓ | ✗ | ✗ | 55.91 | 26.37 | 48.71 | 29.68 | 74.22 | 23.16 | 66.92 | 26.08 |
| ✓ | ✓ | ✗ | <u>55.68</u> | 26.43 | <u>47.76</u> | <u>29.70</u> | <u>74.01</u> | 23.32 | 65.45 | 26.29 |
| ✓ | ✓ | ✓ | **52.96** | <u>26.64</u> | **46.92** | 29.63 | **70.08** | 23.48 | **59.26** | 26.12 |

*Table 3.* Ablation studies on the design components

| $\sigma_y$ | $\Gamma$ | PSNR | FID |
|---|---|---|---|
| 0.0 | glue | 26.51 | 54.69 |
| | Ours | **26.80** | **54.58** |
| 0.01 | glue | 26.39 | 56.47 |
| | Ours | **26.43** | **55.68** |
| 0.05 | glue | 23.86 | 68.99 |
| | Ours | **24.92** | **65.90** |

*Table 4.* Choice of $\Gamma$

approach is also useful for targetting arbitrary resolution image restoration, as the errors accumulated by processing latents in higher dimensions can be corrected through our projection approach. Remarkably, we see that our approach often offers better results (e.g. see Fig. 5) than operating in strided patches (Bar-Tal et al., 2023; Jiménez, 2023), which requires quadratic scaling of compute time.

**Choice of $\mathbf{\Gamma}$** When projecting to the range space of $\mathcal{E}$, we choose to use the proximal optimization strategy in Eq. (19) and Eq. (20). Instead, one could resort to projection to the measurement subspace ("gluing" of (Rout et al., 2023b)) by using $\mathbf{\Gamma}(\hat{\boldsymbol{x}}_0) = \boldsymbol{A}^\top \boldsymbol{y} + (\boldsymbol{I} - \boldsymbol{A}^\top \boldsymbol{A})\hat{\boldsymbol{x}}_0$. In Table 4, we compare our choice of $\mathbf{\Gamma}$ against the gluing on various noise levels on FFHQ SR$\times$8. We see that for all noise levels, our projecton steps consistently outperform the gluing, even when $\mathbf{\Gamma}$ is applied every $\gamma = 4$ steps of reverse diffusion. Furthermore, the differences become more pronounced as we increase the noise level. The difference in the compute time between the two choices is minimal: 331.7 [s] vs 333.2 [s] measured in wall-clock time using RTX 3090 GPU per the restoration of a single image when we compare gluing vs. proximal optimization.

**Visualization of the optimized prompt** Although the optimized prompt during the P2L inference cannot be directly decoded as a text, we can indirectly try to visualize what the prompt has *learned* from the optimization process. If the embedding was optimized in a meaningful way, we would expect it to contain some information about the underlying image. Hence, when we use this embedding to generate samples with standard CFG, we would achieve images that are more similar to the underlying image, compared to not using this embedding. In Fig. 12, we verify that this is indeed the case on the SR$\times$8 experiment on AFHQ cat and dog images.

## 5. Conclusion

We proposed P2L, a latent diffusion model-based inverse problem solver that introduces two new strategies. First, a prompt tuning method to optimize the continuous input text embedding used for DMs was developed. We observed that our strategy can boost the performance by a good margin compared to the usage of null text embedding that prior works employ. Second, a projection approach to keep the latents in the range space of the encoder during the reverse diffusion process was proposed. We show that our approach paves way to jointly utilizing diffusion generative prior and the VAE generative prior. Our approach effectively mitigated the artifacts that often arise during inverse problem solving, while also sharpening the final output. P2L outperforms previous diffusion model-based inverse problem solvers that operate on the latent and the image domain.

**Limitations** While prompt tuning enhances the performance, it also incurs additional computational complexity as additional forward/backward passes through the latent diffusion model and the decoder is necessary. Consequently, the method will need future investigations when aiming for time-critical applications. As we optimize the continuous text embeddings rather than the discrete text directly, it is hard to decipher what the text embedding after the optimization has converged to explicitly. This is a limitation of the text embedder used for SD, as CLIP does not utilize a decoder. We could instead opt for the use of Imagen (Saharia et al., 2022b), where T-5 with an encoder-decoder architecture is used, where one could easily check the learned text from our prompt-tuning scheme.

## Impact Statement

This paper presents work whose goal is to advance inverse problem solving through generative modeling. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## Acknowledgements

We are grateful to José Lezama and Jason Baldridge for their valuable feedback. We also extend our gratitude to Shlomi Fruchter, Kevin Murphy, Mohammad Babaeizadeh, and Han Zhang for their instrumental contributions in facilitating the implementation of the latent diffusion models.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT, Ministry of Science and ICT) (No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation), by National Research foundation of Korea(NRF) (**RS-2023-00262527**), by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)), and by the National Research Foundation of Korea under Grant RS-2024-00336454

## References

Balke, T., Davis, F., Garcia-Cardona, C., McCann, M., Pfister, L., and Wohlberg, B. Scientific Computational Imaging COde (SCICO). Software library available from https://github.com/lanl/scico, 2022.

Bar-Tal, O., Yariv, L., Lipman, Y., and Dekel, T. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43, 2018.

Blau, Y. and Michaeli, T. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6228–6237, 2018.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. Compressed sensing using generative models. In *International conference on machine learning*, pp. 537–546. PMLR, 2017.

Boyd, S., Parikh, N., and Chu, E. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Chung, H. and Ye, J. C. Score-based diffusion models for accelerated mri. *Medical Image Analysis*, pp. 102479, 2022.

Chung, H., Kim, J., Kim, S., and Ye, J. C. Parallel diffusion models of operator and image for blind inverse problems. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023a.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Chung, H., Ryu, D., Mccann, M. T., Klasky, M. L., and Ye, J. C. Solving 3d inverse problems using pre-trained 2d diffusion models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023c.

Chung, H., Lee, S., and Ye, J. C. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *International Conference on Learning Representations*, 2024.

Daras, G., Dagan, Y., Dimakis, A. G., and Daskalakis, C. Score-guided intermediate layer optimization: Fast langevin mixing for inverse problem. *arXiv preprint arXiv:2206.09104*, 2022.

Delbracio, M. and Milanfar, P. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *arXiv preprint arXiv:2303.11435*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dhariwal, P. and Nichol, A. Q. Diffusion models beat GANs on image synthesis. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Efron, B. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Feng, B. T., Smith, J., Rubinstein, M., Chang, H., Bouman, K. L., and Freeman, W. T. Score-based diffusion models as principled priors for inverse imaging. *arXiv preprint arXiv:2304.11751*, 2023.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

González, M., Almansa, A., and Tan, P. Solving inverse problems by joint posterior maximization with autoencoding prior. *SIAM Journal on Imaging Sciences*, 15(2): 822–859, 2022.

He, L., Yan, H., Luo, M., Luo, K., Wang, W., Du, W., Chen, H., Yang, H., and Zhang, Y. Iterative reconstruction based on latent diffusion model for sparse data reconstruction. *arXiv preprint arXiv:2307.12070*, 2023.

Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., and Cohen-Or, D. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Jiménez, Á. B. Mixture of diffusers for scene composition and high resolution image generation. *arXiv preprint arXiv:2302.02412*, 2023.

Kadkhodaie, Z. and Simoncelli, E. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13242–13254. Curran Associates, Inc., 2021.

Kamilov, U. S., Bouman, C. A., Buzzard, G. T., and Wohlberg, B. Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 40(1):85–97, 2023.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

Kawar, B., Vaksman, G., and Elad, M. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=kxXvopt9pWK.

Kim, J., Park, G. Y., Chung, H., and Ye, J. C. Regularization by texts for latent diffusion inverse solvers. *arXiv preprint arXiv:2311.15658*, 2023.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Laroche, C., Almansa, A., and Coupete, E. Fast diffusion em: a diffusion model for blind inverse problems with application to deconvolution. *arXiv preprint arXiv:2309.00287*, 2023.

Lee, S., Chung, H., Park, M., Park, J., Ryu, W.-S., and Ye, J. C. Improving 3D imaging with pre-trained perpendicular 2D diffusion models. *arXiv preprint arXiv:2303.08440*, 2023.

Mardani, M., Song, J., Kautz, J., and Vahdat, A. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023.

Mokady, R., Hertz, A., Aberman, K., Pritch, Y., and Cohen-Or, D. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.

Murata, N., Saito, K., Lai, C.-H., Takida, Y., Uesaka, T., Mitsufuji, Y., and Ermon, S. Gibbsddrm: A partially collapsed gibbs sampler for solving blind inverse problems with denoising diffusion restoration. *arXiv preprint arXiv:2301.12686*, 2023.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FjNys5c7VyY.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Romano, Y., Elad, M., and Milanfar, P. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Rout, L., Chen, Y., Kumar, A., Caramanis, C., Shakkottai, S., and Chu, W.-S. Beyond first-order tweedie: Solving inverse problems using latent diffusion. *arXiv preprint arXiv:2312.00852*, 2023a.

Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A. G., and Shakkottai, S. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *arXiv preprint arXiv:2307.00619*, 2023b.

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022a.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022b.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, B., Kwon, S. M., Zhang, Z., Hu, X., Qu, Q., and Shen, L. Solving inverse problems with latent diffusion models via hard data consistency. *arXiv preprint arXiv:2307.08123*, 2023a.

Song, J., Vahdat, A., Mardani, M., and Kautz, J. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR*, 2021.

Song, Y., Shen, L., Xing, L., and Ermon, S. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vaRCHVj0uGI.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In *2013 IEEE global conference on signal and information processing*, pp. 945–948. IEEE, 2013.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Wang, J., Yue, Z., Zhou, S., Chan, K. C., and Loy, C. C. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023a.

Wang, Y., Yu, J., Yu, R., and Zhang, J. Unlimited-size diffusion restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1160–1167, 2023b.

Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., and Zhu, J. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023c.

Whang, J., Lei, Q., and Dimakis, A. Solving inverse problems with a flow-based noise model. In *International Conference on Machine Learning*, pp. 11146–11157. PMLR, 2021.

Whang, J., Delbracio, M., Talebi, H., Saharia, C., Dimakis, A. G., and Milanfar, P. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16293–16303, 2022.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

Zhu, Y., Zhang, K., Liang, J., Cao, J., Wen, B., Timofte, R., and Van Gool, L. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1219–1229, 2023.

## A. Background on diffusion models

**Lemma A.1** (Tweedie's formula)**.** *Given a Gaussian perturbation kernel $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; s_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$, the posterior mean is given by*

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \frac{1}{\alpha_t}(\boldsymbol{x}_t + \sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)) \tag{21}$$

*Proof.*

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t) &= \frac{\nabla_{\boldsymbol{x}_t} p(\boldsymbol{x}_t)}{p(\boldsymbol{x}_t)} && (22)\\
&= \frac{1}{p(\boldsymbol{x}_t)}\nabla_{\boldsymbol{x}_t}\int p(\boldsymbol{x}_t|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && (23)\\
&= \frac{1}{p(\boldsymbol{x}_t)}\int \nabla_{\boldsymbol{x}_t} p(\boldsymbol{x}_t|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && (24)\\
&= \frac{1}{p(\boldsymbol{x}_t)}\int p(\boldsymbol{x}_t|\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && (25)\\
&= \int p(\boldsymbol{x}_0|\boldsymbol{x}_t)\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && (26)\\
&= \int p(\boldsymbol{x}_0|\boldsymbol{x}_t)\frac{s_t\boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2}\,d\boldsymbol{x}_0 && (27)\\
&= \frac{s_t\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] - \boldsymbol{x}_t}{\sigma_t^2}. && (28)
\end{aligned}$$

Rearranging the terms, we achieve the conclusion. □

Lemma A.1 lets us compute the posterior mean when we have access to the score function. In diffusion models, we parametrize the score function with a neural network and train it through denoising score matching

$$\theta^* = \arg\min_{\theta}\mathbb{E}_{t\sim U[0,1],\boldsymbol{x}_0\sim p_{\text{data}},\boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{I})}\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) - \nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2. \tag{29}$$

Let us consider the case of DDPM (Ho et al., 2020) with the forward perturbation kernel $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I})$[8]. Then, we have the following alternative parametrizations

$$\boldsymbol{s}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t,t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t,t) = \frac{\sqrt{\bar{\alpha}_t}D_{\theta^*}(\boldsymbol{x}_0) - \boldsymbol{x}_t}{\sqrt{1-\bar{\alpha}_t}}, \tag{30}$$

where the second parametrization comes from epsilon-matching (Ho et al., 2020) and is mostly used throughout the work, and the last parametrization directly estimates the posterior mean by regarding the diffusion model as a denoiser.

**Corollary A.2** (Conditional Tweedie's formula)**.**

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y}] = \frac{1}{s_t}(\boldsymbol{x}_t + \sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{y})) \tag{31}$$

The corollary is a simple consequence of conditioning the Tweedie's formula with an additional variable $\boldsymbol{y}$. As $\log p(\boldsymbol{x}_t|\boldsymbol{y})$ is intractable, we can estimate Eq. (31), with the choices of $s_t, \sigma_t$ made from DDPM, with (Chung et al., 2023b)

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y}] &= \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t}(\log p(\boldsymbol{x}_t) + \log p(\boldsymbol{y}|\boldsymbol{x}_t))) && (32)\\
&\overset{\text{(DPS)}}{\approx} \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\left(\boldsymbol{s}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t,t) + \nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0)\right)) && (33)\\
&= \hat{\boldsymbol{x}}_0 + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}}\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0) && (34)
\end{aligned}$$

[8] In the discrete setup, $\bar{\alpha}_t := \prod_{i=1}^t \alpha_t$, and $\alpha_t := 1-\beta_t$ with $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I})$

where $\hat{\boldsymbol{x}}_0 := D_{\theta^*}(\boldsymbol{x}_t, t)$. Further, we can circumvent the need to backpropagate through the diffusion model and save computation by using the DDS approximation (Chung et al., 2024)

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] \overset{(\text{DDS})}{\approx} \hat{\boldsymbol{x}}_0 + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} \nabla_{\hat{\boldsymbol{x}}_0} \log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0), \tag{35}$$

where the difference stems from that we take the gradient w.r.t. $\hat{\boldsymbol{x}}_0$ rather than $\boldsymbol{x}_t$. Running Eq. (4) with the approximations Eq. (33) or Eq. (35) amounts to approximately sampling from the posterior distribution.

## B. Proof-of-concept experiment

For the caption generation with PALI, we simply take the captions with the highest score. Examples of the captions generated from PALI are presented in Fig. 9. In our initial experiments, we found that using PALI captions directly did not directly lead to an improvement in the performance, as it only describes the *content* of the image, and says nothing about the *quality* of the image. Therefore, we use the following text prompts for the oracle "`A high quality photo of a` {PALI_prompt}", similar to the general text prompts.

For both inverse problems (SR$\times$8, inpainting with $p = 0.8$), we use the LDPS algorithm with 1000 NFE and $\eta = 0.0$. We apply prompt tuning algorithm per denoising step as indicated in Algorithm 2, with $K = 5$ and learning rate of $1e-4$. When optimizing for the text embedding, we initialize it with the embedding vector from the token "`A high quality photo of a face`" for FFHQ, and "`A high quality photo`" for ImageNet in the case of inpainting. Note that for the latter, we did not find much performance difference when initializing from the null text prompt, or even initializing it with "`A high quality photo of a dog`". For $\times 8$ SR, we initialize the text embeddings from PALI captions generated from $\boldsymbol{y}$, as we empirically observe that PALI captions from $\boldsymbol{y}$ still have a relatively good coarse description about the given image.

## C. Runtime analysis

In Tab. 5, we include the runtime for each algorithm used in the paper when solving inverse problems with diffusion models, measured in wall-clock time [s] with a single RTX 3090 GPU. Note that P2L ($K = 0$) corresponds to the case where we do not use prompt-tuning, and only apply the idea of leveraging the VAE prior (i.e. encoder range space projection). In this case, the compute time is roughly equivalent to the LDIS baselines. As we increase the number of iterations for prompt embedding optimization, the required computation time approximately linearly increases. In this regard, P2L requires more compute against other LDIS baselines as we additionally optimize for the text prompt, which can be considered a downside of the approach. However, it should be noted that P2L is the first approach that shows the possibility and feasibility of the approach. While it may not be computationally efficient at this point, P2L would be a good cornerstone that future works can build upon to devise faster, more efficient solvers.

| Method | Time [s] | FID↓ | PSNR↑ | Type |
|---|---|---|---|---|
| P2L ($K = 5$) | 1982.7 | 51.81 | 23.38 | Latent diffusion |
| P2L ($K = 3$) | 1333.6 | 52.90 | 23.36 | |
| P2L ($K = 1$) | 657.3 | 55.62 | 23.35 | |
| P2L ($K = 0$) | 333.2 | 56.20 | 23.30 | |
| LDPS | 313.9 | 61.09 | 23.21 | |
| GML-DPS (Rout et al., 2023b) | 390.6 | 60.36 | 23.21 | |
| PSLD (Rout et al., 2023b) | 408.7 | 60.81 | 23.17 | |
| LDIR (He et al., 2023) | 317.2 | 63.46 | 22.23 | |
| DDS (Chung et al., 2024) | 20.1 | 203.2 | 12.72 | Pixel diffusion |
| DPS (Chung et al., 2023b) | 291.0 | 54.61 | 20.70 | |
| DiffPIR (Zhu et al., 2023) | 21.2 | 488.3 | 13.44 | |
| ΠGDM (Song et al., 2023b) | 30.2 | 53.00 | 21.08 | |

*Table 5.* Comparison in compute time for each method using RTX 3090 GPU in wall-clock time [s].

## D. Implementation details

### D.1. $\mathcal{C}$ update prompt tuning

We consider the following optimization problem

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}\left(\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}]\right)\|_2^2, \tag{36}$$

| | FFHQ | | | | ImageNet | | | |
|---|---|---|---|---|---|---|---|---|
| problem | Deblur (motion) | Deblur (gauss) | SR×8 | inpaint | Deblur (motion) | Deblur (gauss) | SR×8 | inpaint |
| Gradient type | Adam | Adam | GD | Adam | Adam | GD | GD | GD |
| $\rho_t$ | 0.05 | 0.05 | 1.0 | 0.05 | 0.1 | $\bar{\alpha}_t$ | $15\bar{\alpha}_t$ | 0.5 |
| $\gamma$ | 5 | 4 | 4 | 3 | 5 | 4 | 4 | 3 |
| $\lambda$ | 1.0 | 1.0 | 1.0 | 0.1 | 1.0 | 1.0 | 1.0 | 0.1 |
| $K$ | 3 | 5 | 5 | 1 | 3 | 3 | 3 | 1 |
| learning rate | $5e-5$ | $1e-4$ | $1e-4$ | $1e-4$ | $1e-5$ | $1e-4$ | $1e-5$ | $1e-4$ |

*Table 6.* Hyper-parameter choice for the proposed method. White shade: hyper-parameters related to gradient updates, blue shade: hyper-parameters related to projecting onto the range space of $\mathcal{E}$, red shade: hyper-parameters related to prompt tuning.

where Eq. (36) is performed for every timestep $t$ during the inference stage. Here, we approximate the conditional posterior mean as

$$\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t,\boldsymbol{y},\mathcal{C}] = \frac{1}{\sqrt{\bar{\alpha}_t}}\boldsymbol{z}_t + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}}\left(\nabla_{\boldsymbol{z}_t}\log p(\boldsymbol{z}_t|\mathcal{C}) + \nabla_{\boldsymbol{z}_t}\log p(\boldsymbol{y}|\boldsymbol{z}_t,\mathcal{C})\right) \tag{37}$$

$$\simeq \hat{\boldsymbol{z}}_0^{(\mathcal{C})} + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}}\nabla_{\boldsymbol{z}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{z}}_0^{(\mathcal{C})}) \tag{38}$$

$$\simeq \hat{\boldsymbol{z}}_0^{(\mathcal{C})} + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}}\nabla_{\hat{\boldsymbol{z}}_0^{(\mathcal{C})}}\log p(\boldsymbol{y}|\hat{\boldsymbol{z}}_0^{(\mathcal{C})}), \tag{39}$$

which is the consequence of the DDS approximation in Eq. (35). Notice that we update our embeddings to improve the fidelity Eq. (36). However, in practice, this also leads to higher quality images in terms of perception. For optimizing Eq. (36), we use Adam with the learning rate and the number of iterations as denoted in Table 6 for every $t$. In practice, we choose a static step size $\rho = 1.0$ with the gradient of the norm, which was shown to be effective in (Chung et al., 2023b). The resulting prompt tuning algorithm is summarized in Algorithm 2.

**Algorithm 3** P2L: Adam

**Require:** $\boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}, \boldsymbol{z}_T, \boldsymbol{y}, \mathcal{C}, T, K, \gamma, \beta_1, \beta_2, \varepsilon, \boldsymbol{\Gamma}$
$\boldsymbol{m}_T \leftarrow$ `np.zeros_like`$(\boldsymbol{z}_T)$
$\boldsymbol{v}_T \leftarrow$ `np.zeros_like`$(\boldsymbol{z}_T)$
**for** $t = T$ **to** $1$ **do**
  $\mathcal{C}_t^* \leftarrow \textsc{OptimizeEmb}(\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}_t^0, K)$
  $\hat{\boldsymbol{\epsilon}}_t \leftarrow \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\boldsymbol{z}_t, \mathcal{C}_t^*)$
  $\hat{\boldsymbol{z}}_{0|t} \leftarrow (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)/\sqrt{\bar{\alpha}_t}$
  **if** $(t \mod \gamma) = 0$ **then**
    $\hat{\boldsymbol{z}}'_{0|t} \leftarrow \mathcal{E}\left(\boldsymbol{\Gamma}\left(\mathcal{D}(\hat{\boldsymbol{z}}_{0|t})\right)\right)$
  **end if**
  $\boldsymbol{z}'_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}'_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_t$
  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{z}_t}\|\mathcal{A}\mathcal{D}(\hat{\boldsymbol{z}}_{0|t}) - \boldsymbol{y}\|$
  $\hat{\boldsymbol{m}}_{t-1} \leftarrow (\beta_1\boldsymbol{m}_t + (1-\beta_1)\boldsymbol{g})/(1-\beta_1)$
  $\hat{\boldsymbol{v}}_{t-1} \leftarrow (\beta_2\boldsymbol{v}_t + (1-\beta_2)(\boldsymbol{g}\circ\boldsymbol{g}))/(1-\beta_2)$
  $\boldsymbol{z}_{t-1} \leftarrow \boldsymbol{z}'_{t-1} - \rho_t\frac{\hat{\boldsymbol{m}}_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_{t-1}+\varepsilon}}$
  $\mathcal{C}_{t-1}^{(0)} \leftarrow \mathcal{C}_t^*$
**end for**
**return** $\boldsymbol{x}_0 \leftarrow \mathcal{D}(\boldsymbol{z}_0)$

### D.2. $\boldsymbol{z}_t$ update

In Table 6, there are two gradient types: GD and Adam. For GD, we use standard gradient descent steps as presented in Algorithm 1. For Adam, using the same prompt tuning Algorithm 2, we adopt a history gradient update scheme as

proposed in (He et al., 2023) to arrive at Algorithm 3. Note that the hyper-parameters of the Adam update were fixed to be $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1e-8$, which is the default setting. We only search for the optimal step size $\rho_t$ via grid search, which is set to 0.1 for motion deblurring in ImageNet, and 0.05 otherwise.

### D.3. Comparison methods

**LDPS** LDPS in Eq. (6) can be considered a straightforward extension image domain DPS (Chung et al., 2023b). The three works that we review in this section (He et al., 2023; Rout et al., 2023b; Song et al., 2023a) all consider LDPS as a baseline. In Eq. (6), we use a static step size of $\rho = 1$, widely adopted in literature.

**LDIR (He et al., 2023)** Using Adam-like history gradient update scheme, a single iteration of the algorithm can be summarized as follows

$$\boldsymbol{g}_t = \nabla_{\boldsymbol{z}_t} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}(\hat{\boldsymbol{z}}_0)\| \tag{40}$$

$$\hat{\boldsymbol{m}}_t = (\beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t)/(1-\beta_1) \tag{41}$$

$$\hat{\boldsymbol{v}}_t = (\beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)(\boldsymbol{g}_t \circ \boldsymbol{g}_t))/(1-\beta_2) \tag{42}$$

$$\boldsymbol{z}_{t-1} = \mathrm{DDIM}(\boldsymbol{z}_t) - \rho \frac{\hat{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t} + \varepsilon}, \tag{43}$$

where $\circ$ denotes element-wise product, and $\beta_1, \beta_2, \varepsilon$ are the hyperparameters of the sampling scheme. As LDIR uses a momentum-based update scheme, we have smoother gradient transitions. We fix $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1e-8$ to be identical to when using the proposed method. The step size $\rho$ is chosen to be the optimal value found through grid search: 0.1 for ImageNet motion deblurring, and 0.05 otherwise.

**GML-DPS, PSLD (Rout et al., 2023b)** GML-DPS attempts to regularize the predicted clean latent $\hat{\boldsymbol{z}}_0$ to be a fixed point after encoding and decoding. Formally, the update step reads

$$\boldsymbol{z}_{t-1} = \mathrm{DDIM}(\boldsymbol{z}_t) - \rho \nabla_{\boldsymbol{z}_t} \left( \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}(\hat{\boldsymbol{z}}_0)\|_2 + \gamma \|\hat{\boldsymbol{z}}_0 - \mathcal{E}(\mathcal{D}(\hat{\boldsymbol{z}}_0))\|_2 \right). \tag{44}$$

Further, PSLD applies an orthogonal projection onto the subspace of $\boldsymbol{A}$ in between decoding and encoding to enforce fidelity

$$\boldsymbol{z}_{t-1} = \mathrm{DDIM}(\boldsymbol{z}_t) - \rho \nabla_{\boldsymbol{z}_t} \left( \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}(\hat{\boldsymbol{z}}_0)\|_2 + \gamma \|\hat{\boldsymbol{z}}_0 - \mathcal{E}(\boldsymbol{A}^\top \boldsymbol{y} + (\boldsymbol{I} - \boldsymbol{A}^\top \boldsymbol{A})\mathcal{D}(\hat{\boldsymbol{z}}_0))\|_2 \right). \tag{45}$$

We use the static step size of $\rho = 1$, and choose $\gamma = 0.1$, as advised in (Rout et al., 2023b). GML-DPS and PSLD are closest to the proposed method in spirit, as these methods attempt to guide the latents to stay closer to the natural manifold by enforcing them to be a fixed point after autoencoding. The difference is that these approaches use gradient guidance while we try to explicitly project the latents into the the natural manifold.

**TReg (Kim et al., 2023)** TReg uses CFG with a high guidance scale of 7.5 to produce the denoised estimate $\hat{\boldsymbol{z}}_0$. By defining the CLIP image encoder (Radford et al., 2021) as $\mathcal{T}$ and the cosine similarity function as $\mathrm{sim}(\cdot)$, the algorithm can be represented as

$$\hat{\boldsymbol{x}}_0 = \mathcal{D}(\hat{\boldsymbol{z}}_0) \tag{46}$$

$$\hat{\boldsymbol{x}}_0(\boldsymbol{y}) = \arg\min_{\boldsymbol{x}} \frac{\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x})\|_2^2}{2\sigma^2} + \lambda \|\boldsymbol{x} - \hat{\boldsymbol{x}}_0\|_2^2 \tag{47}$$

$$\hat{\boldsymbol{z}}_0(\boldsymbol{y}) = \mathcal{E}(\hat{\boldsymbol{x}}_0(\boldsymbol{y})) \tag{48}$$

$$\hat{\mathcal{C}}_\varnothing = \mathcal{C}_\varnothing - \eta \nabla_{\mathcal{C}_\varnothing} \mathrm{sim}(\mathcal{T}(\hat{\boldsymbol{x}}_0(\boldsymbol{y})), \mathcal{C}_\varnothing) \tag{49}$$

$$\boldsymbol{z}_{t-1} = \mathrm{DDIM}(\boldsymbol{z}_t) - \rho_t \nabla_{\boldsymbol{z}_t} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}(\hat{\boldsymbol{z}}_0, \mathcal{C}_\varnothing)\|_2^2 \tag{50}$$

The crucial difference of TReg is that it makes updates with respect to the null text embedding with high CFG scale, which greatly emphasizes the text conditioning, while P2L makes updates on the conditional text embeddings without any CFG.

| steps | 0 | 1 | | | 3 | | | 5 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| lr | - | $1e-5$ | $1e-4$ | $1e-3$ | $1e-5$ | $1e-4$ | $1e-3$ | $1e-5$ | $1e-4$ | $1e-3$ |
| FID | 61.16 | 60.66 | 59.60 | **57.61** | 60.11 | 59.34 | 60.19 | 60.02 | <u>58.59</u> | 62.67 |
| PSNR | 26.49 | 26.69 | 26.71 | 26.73 | **26.78** | 26.70 | 26.61 | <u>26.73</u> | 26.17 | 26.38 |

*Table 7.* Robustness to hyper-parameters in prompt-tuning. FFHQ SR×8 on 256 test images. **Bold**: best, <u>underline</u>: second best.

**DPS (Chung et al., 2023b)** DPS is a DIS that utilizes the following update scheme[9]

$$\boldsymbol{x}_{t-1} = \mathrm{DDIM}(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t}\left(\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_2\right). \tag{51}$$

The optimal value of $\eta$ was found through grid search for each inverse problem: $\eta = 0.0$ for SR×8, and $\eta = 1.0$ for others.

**ΠGDM (Song et al., 2023b)** Similar to DPS, ΠGDM considers the following gradient update scheme

$$\boldsymbol{x}_{t-1} = \mathrm{DDIM}(\boldsymbol{x}_t) - \left((\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0)^\top (r_t^2 \boldsymbol{A}\boldsymbol{A}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{A} \frac{\partial \hat{\boldsymbol{x}}_0}{\partial \boldsymbol{x}_t}\right)^\top, \tag{52}$$

where $r_t$ is a hyper-parameter and $\sigma$ is the noise level of the measurement. We take $r_t$ as advised in (Song et al., 2023b), and use 100 step DDIM sampling with $\eta = 1.0$ for all experiments.

**DDS (Chung et al., 2024)** The following updates are used

$$\hat{\boldsymbol{x}}_0' = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \frac{\gamma}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_0\|_2^2 \tag{53}$$

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_0' + \sqrt{1 - \bar{\alpha}_{t-1} - \eta^2\tilde{\beta}_{t-1}^2}\hat{\boldsymbol{\epsilon}}_t + \eta\tilde{\beta}_{t-1}\boldsymbol{\epsilon}, \tag{54}$$

where Eq. (53) is solved through CG with 5 iterations, $\gamma = 1.0$. $\eta = 0.0$ is chosen for Gaussian deblurring, and $\eta = 1.0$ for the rest of the inverse problems.

**DiffPIR (Zhu et al., 2023)** Similar to DDS, the following updates are used

$$\hat{\boldsymbol{x}}_0' = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \frac{\lambda\sigma^2\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_0\|_2^2 \tag{55}$$

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_0' + \sqrt{1 - \bar{\alpha}_{t-1}}(\sqrt{1-\zeta}\hat{\boldsymbol{\epsilon}}_t + \sqrt{\zeta}\boldsymbol{\epsilon}), \tag{56}$$

where $\sigma$ is the noise level of the measurement, and $\lambda, \zeta$ are hyper-parameters. Unlike DDS, the solution to Eq. (55) is obtained as a closed-form solution. These hyper-parameters are found through grid search. SR×8: $\zeta = 0.35, \lambda = 35.0$ / Deblur: $\zeta = 0.3, \lambda = 7.0$ / Inpaint: $\zeta = 1.0/\lambda = 7.0$.

## E. Efficient implementation in JAX

In model-based inverse problem solving, having access to efficient computation of the adjoint $\boldsymbol{A}^\top$ is a must. Here, we consider a general case of solving linear inverse problems where the computation of SVD is too costly, and hence one has to define the adjoint operator manually (e.g. computed tomography). Furthermore, for cases such as deblurring from circular convolution, one needs to carefully design the operator, as there are many potential pitfalls (e.g. boundary, size mismatch). These are more often than not the limiting factors of the applicability of the model-based approaches for solving inverse problems. We show in Fig. 4 that this can be much alleviated by using `jax`, as we can implicitly define a transpose operator with reverse-mode automatic differentiation (Baydin et al., 2018). We note this design was also established in (Balke et al., 2022).

[9]The original work only considered DDPM sampling. We consider DDIM as a generalization of DDPM as it can be retrieved with $\eta = 1.0$.

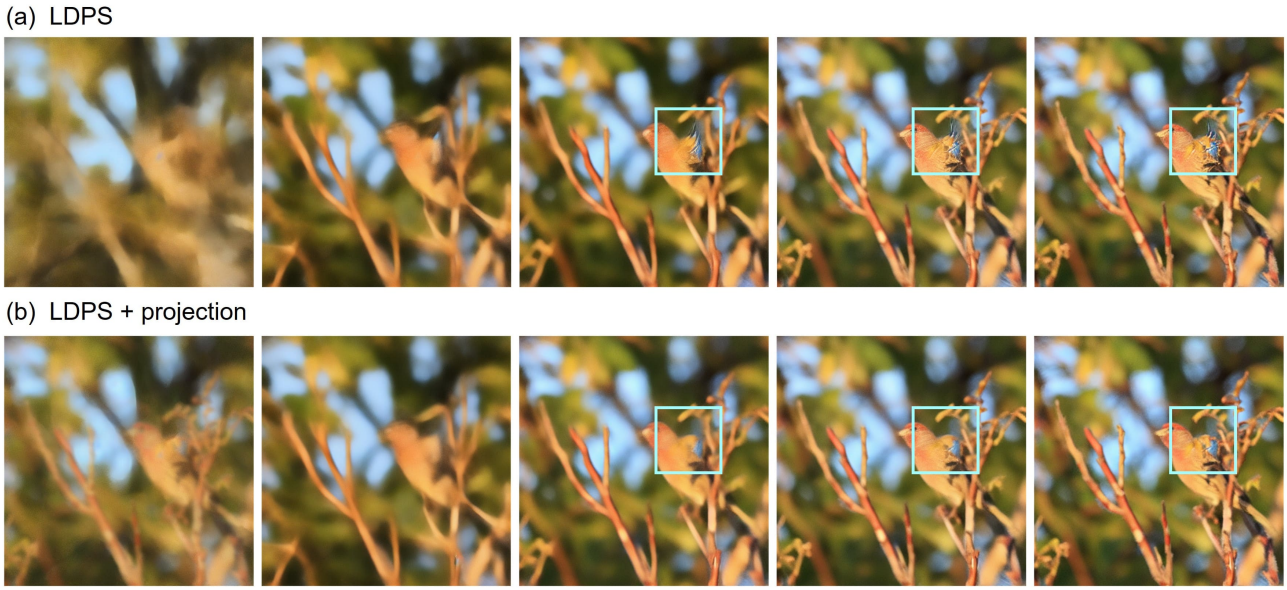


*Figure 2.* Evolution of DIS while solving SR×8 with (a) LDPS, (b) LDPS + projection. Using projection steps help mitigate the artifacts.

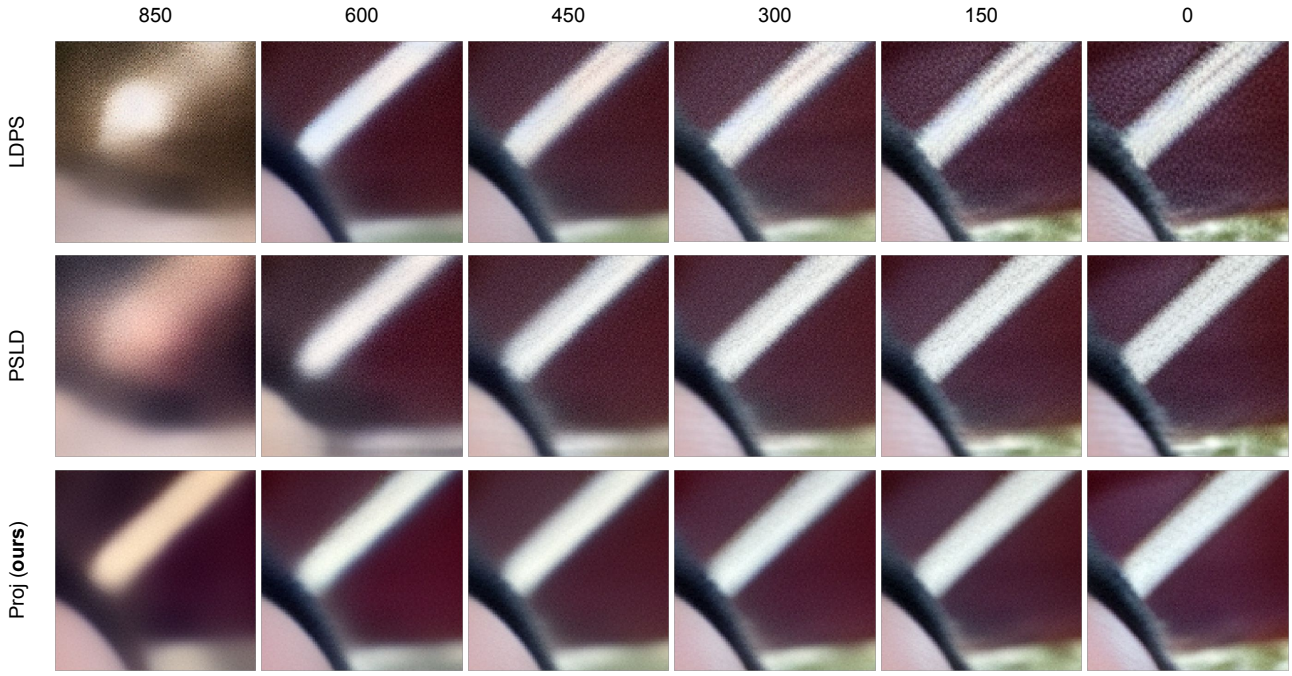


*Figure 3.* Close-up of the progress of $\mathcal{D}(\hat{\boldsymbol{z}}_0)$ through time $t$ when solving $\times 8$ SR on FFHQ.

## F. Targetting arbitrary resolution

For SD, using an encoder to convert from the image to the latent space reduces the dimension by $\times 8$. When training SD, the diffusion model that operates on the latent space was trained with 64×64 latents, obtained from 512×512 images. When the image that we wish to restore (or generate) is larger than 512×512, the latents will also be larger than 64×64. In this case, due to the train-test time discrepancy, the results that we get will be suboptimal if one processes the larger latent as a

```
ones = jnp.ones(x.shape)
_, _AT = jax.vjp(A_funcs.A, ones)
AT = lambda y: _AT(y)[0]
A_funcs.AT = AT
def cg_A(x, cg_lamb):
    return A_funcs.AT(A_funcs.A(x)) + cg_lamb * x
hatx0 = D(hatz0)
cg_y = A_funcs.AT(y) + cg_lamb * hatx0
hatx0, _ = jax.scipy.sparse.linalg.cg(cg_A, cg_y, x0=hatx0)
```

*Figure 4.* Defining $\boldsymbol{A}^\top$ can be automatically achieved through `jax.vjp` given that $\boldsymbol{A}$ is differentiable.

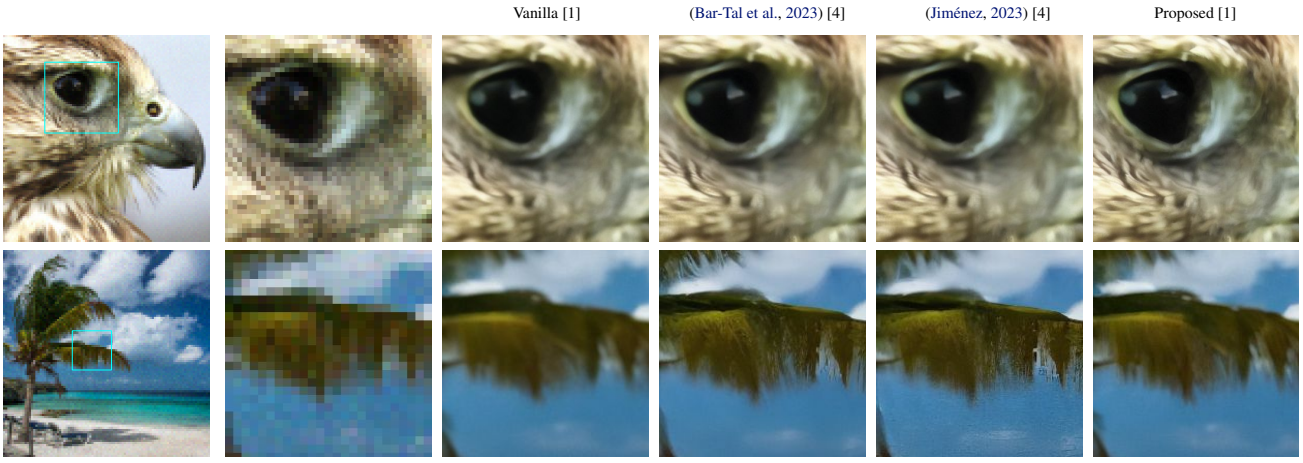


*Figure 5.* Results on $\times 8$ SR on DIV2K validation set of $768\times768$ resolution. [Diffusion NFE per denoising step]. Vanilla and proposed process the latent as a whole.

whole (Fig. 6 (a)). A natural way to counteract this discrepancy is to process the latents in patches[10]. When processing in patches of size $64\times64$ with stride 32 on both directions, it requires us 4 score function NFEs per denoising step (Fig. 6 (c),(d)). (Bar-Tal et al., 2023) uniformly weights the overlapping patches, and (Jiménez, 2023) weights the patches with Gaussian weights with variance 0.01. The downside of these methods is that the number NFEs required for inference scales quadratically with the size of the image.

Notice that all methods that aim for high-resolution synthesis using latent diffusion models only focus on better dealing with the latents and use the decoding part as-is. This is due to the fact that the diffusion models that act in the latent space is more sensitive to the change in the input resolution, and hence the error could easily accumulate if we operate on larger latents directly. On the other hand, VAE is much more robust to the change in the input resolution. When given a latent that stays within the range space of the encoder, the decoder is able to produce a high-quality image directly even when the input size is larger than $64\times$ 64. In this regard, we can project this latent to the range space of $\mathcal{E}$ by setting $\hat{\boldsymbol{z}}_0' = \mathcal{E}(\boldsymbol{\Gamma}(\mathcal{D}(\hat{\boldsymbol{z}}_0)))$ for every few steps, as illustrated in Fig. 6 (b). Even though the proposed method is considerably faster than patch-based methods (Bar-Tal et al., 2023; Jiménez, 2023), we see that one can achieve a comparable, or superior performance, as presented in Fig. 5. Furthermore, in Fig. 7, we show that we can use both patching method and the projection method simultaneously, achieving the best results.

## G. Further experimental results

[10]For all the experiments considered in this paper, we consider $768\times768$ images ($96\times96$ latents).

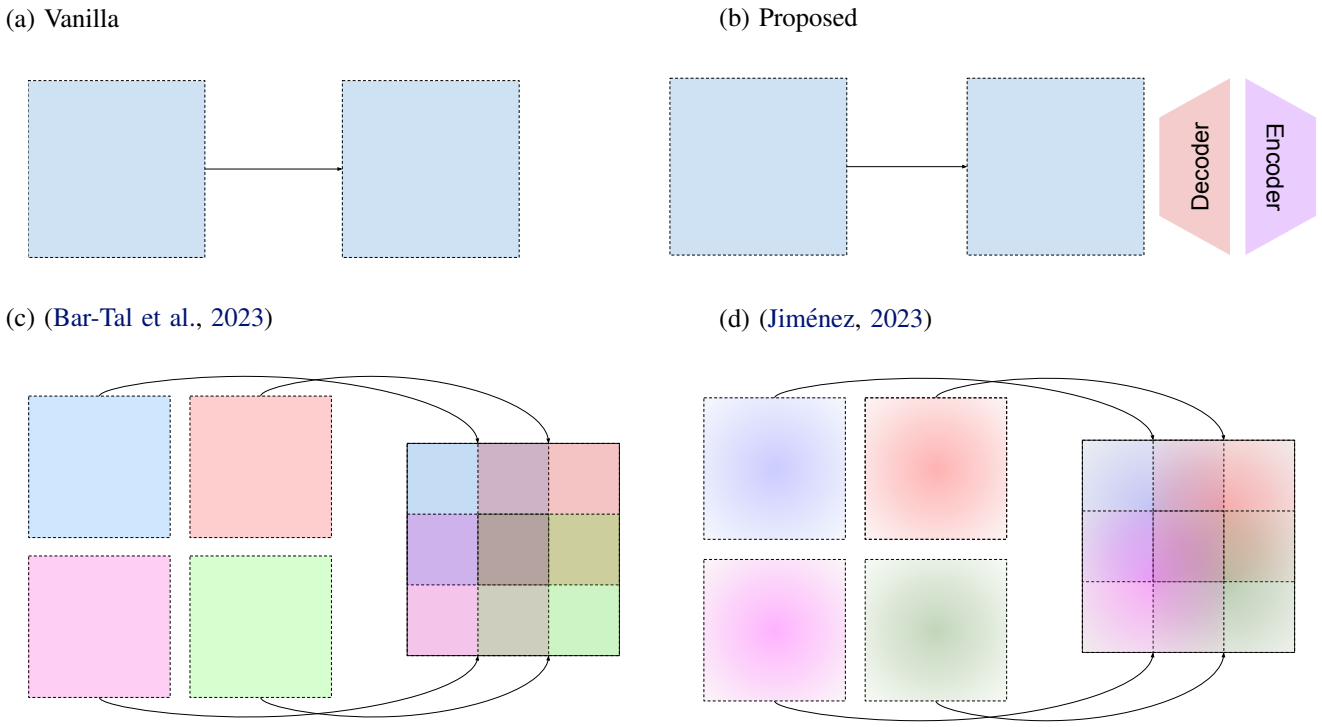


*Figure 6.* Method comparison for processing higher resolution images in the latent space.

| Method | SR (×8) | | | Deblur (motion) | | | Deblur (gauss) | | | Inpaint | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ | FID↓ | LPIPS↓ | PSNR↑ |
| P2L (ours) | **31.23** | **0.290** | 28.55 | 28.34 | **0.302** | **27.23** | **30.62** | **0.299** | 26.97 | **26.27** | **0.168** | 25.29 |
| LDPS | 36.81 | 0.292 | **28.78** | 58.66 | 0.382 | 26.19 | 45.89 | 0.334 | 27.82 | 46.10 | 0.311 | 23.07 |
| GML-DPS (Rout et al., 2023b) | 41.65 | 0.318 | 28.50 | 47.96 | 0.352 | 27.16 | 42.60 | 0.320 | **28.49** | 36.31 | 0.208 | 23.10 |
| PSLD (Rout et al., 2023b) | 36.93 | 0.335 | 26.62 | 47.71 | 0.348 | 27.05 | 41.04 | 0.320 | 28.47 | 35.01 | 0.207 | 23.10 |
| LDIR (He et al., 2023) | 36.04 | 0.345 | 25.79 | **24.40** | 0.376 | 24.40 | 35.61 | 0.341 | 25.75 | 37.23 | 0.250 | **25.47** |
| DDS (Chung et al., 2024) | 262.0 | 1.278 | 13.01 | 88.70 | 1.014 | 14.68 | 74.02 | 0.932 | 17.03 | 113.6 | 0.421 | 17.92 |
| DPS (Chung et al., 2023b) | 47.65 | 0.340 | 21.81 | 65.91 | 0.601 | 21.11 | 100.2 | 0.983 | 15.71 | 137.7 | 0.692 | 15.35 |
| DiffPIR (Zhu et al., 2023) | 141.1 | 1.266 | 13.80 | 72.02 | 0.664 | 21.03 | 69.15 | 0.751 | 22.27 | 33.92 | 0.238 | 24.91 |
| ΠGDM (Song et al., 2023b) | 42.07 | 0.311 | 22.05 | 60.08 | 0.531 | 21.08 | 70.32 | 0.788 | 21.99 | 140.6 | 0.738 | 16.83 |

*Table 8.* Quantitative evaluation (PSNR, LPIPS, FID) of inverse problem solving on FFHQ 512×512-1k validation dataset. **Bold**: best, underline: second best. Methods that are not LDM-based are shaded in gray.

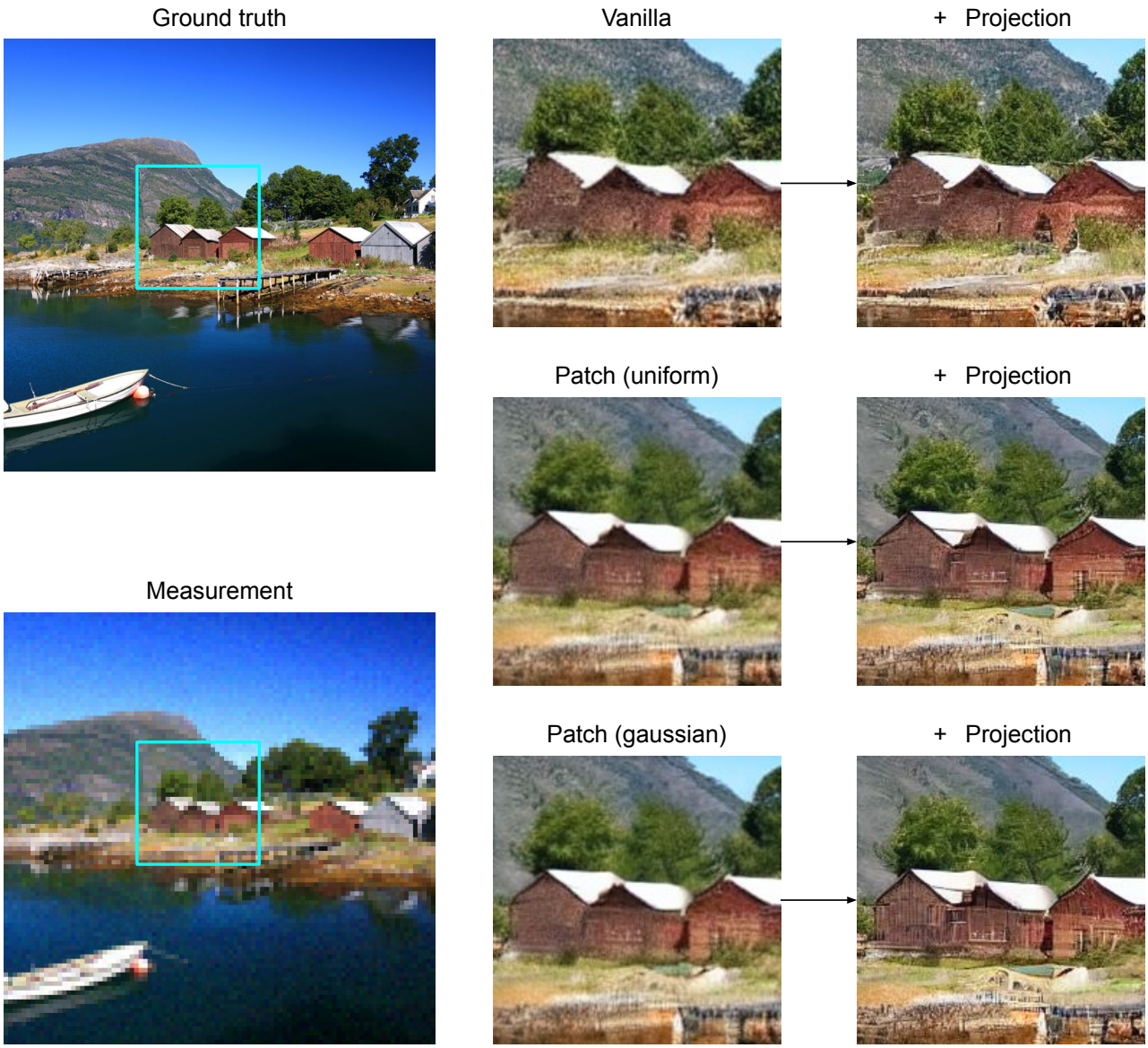


*Figure 7.* Further results on $\times 8$ SR on DIV2K validation set of $768 \times 768$ resolution. Comparison between with and without using our projection approach on various baseline methods.

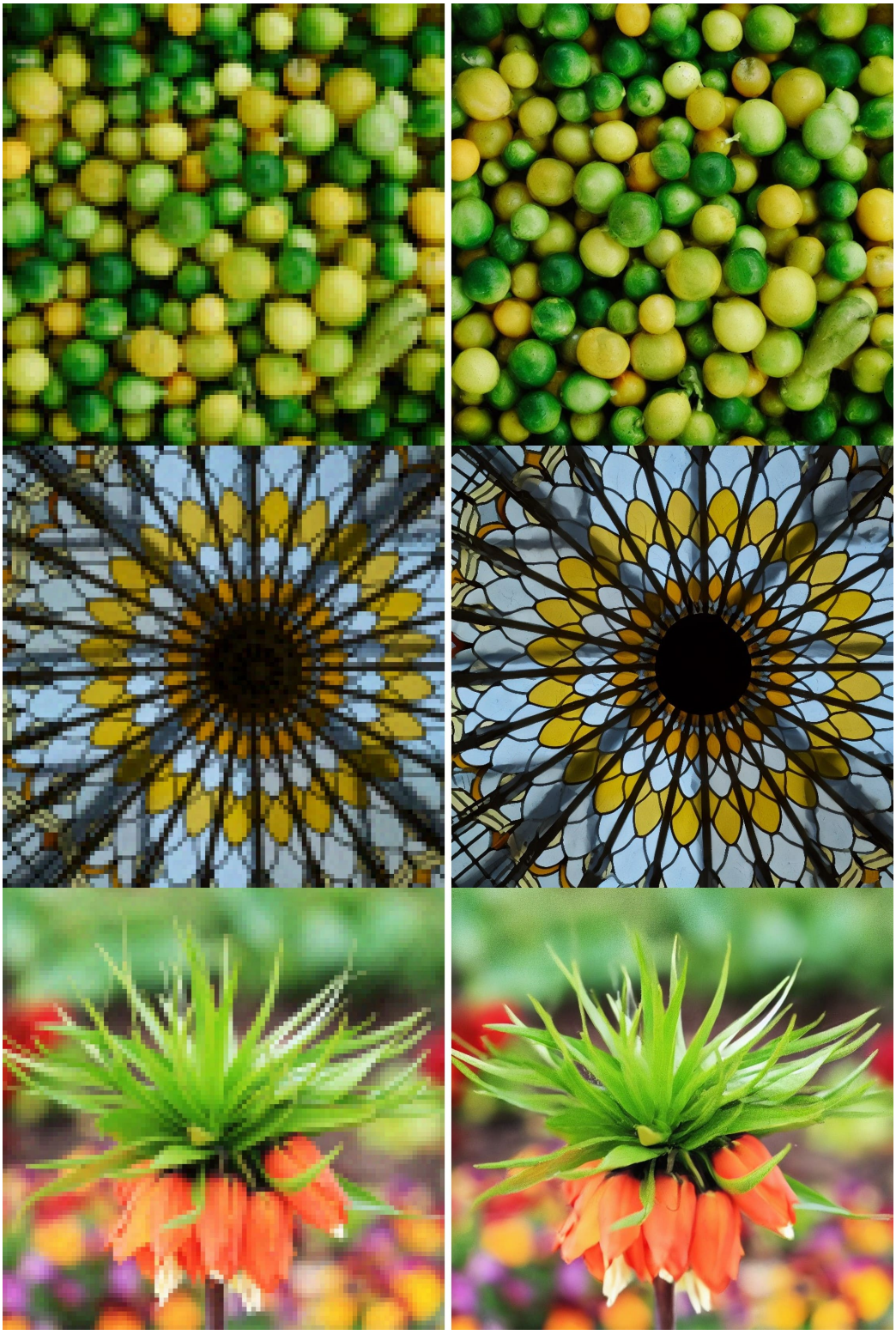

*Figure 8.* Full image results of $\times 8$ SR on DIV2K validation set of $768 \times 768$ resolution. Left: measurement, Right: P2L

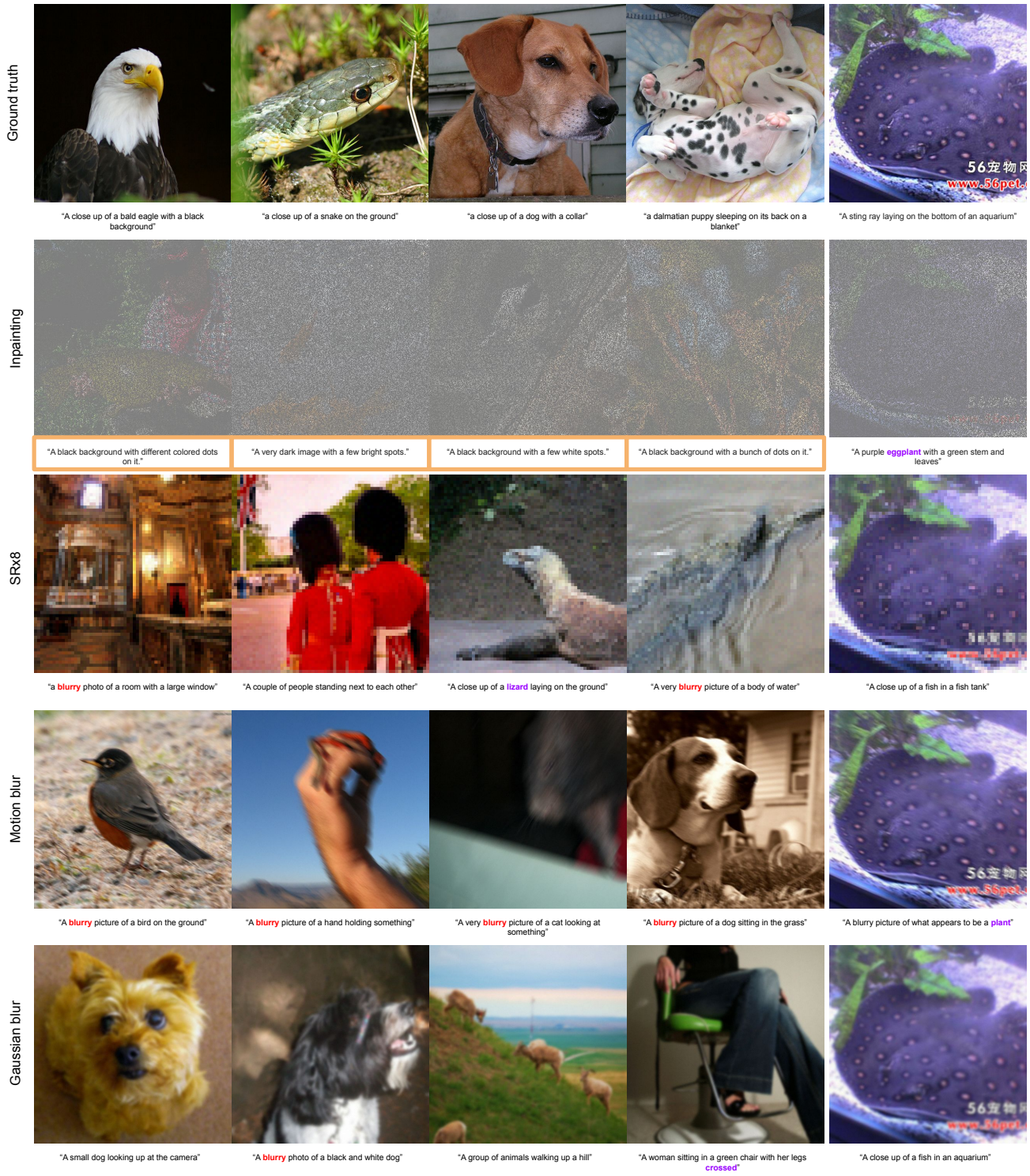


*Figure 9.* Captions generated by PALI (Chen et al., 2022) from ground-truth ImageNet 512×512 clean images, and the degraded images. The rightmost column contain images that are from the same ground truth. Captions in in orange box completely fail to describe the underlying image. Purple captions wrongly identify the image. Captions generated from degraded measurements often contain negative words such as blurry.

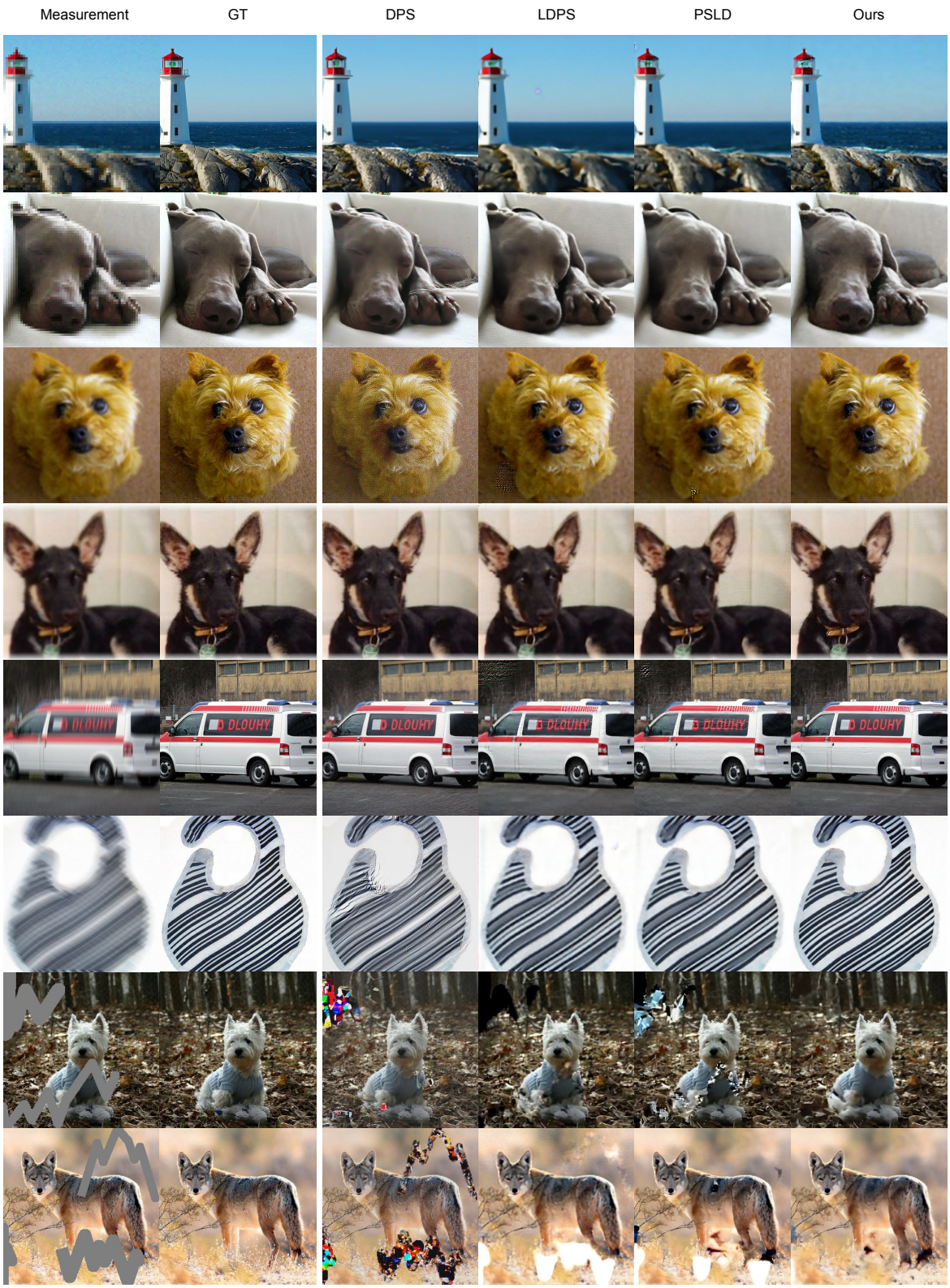


*Figure 10.* ImageNet restoration results. Row 1-2: SR×8, row 3-4: gaussian deblurring, row 5-6: motion deblurring, row 7-8: freeform inpainting; All with $\sigma = 0.01$ noise.

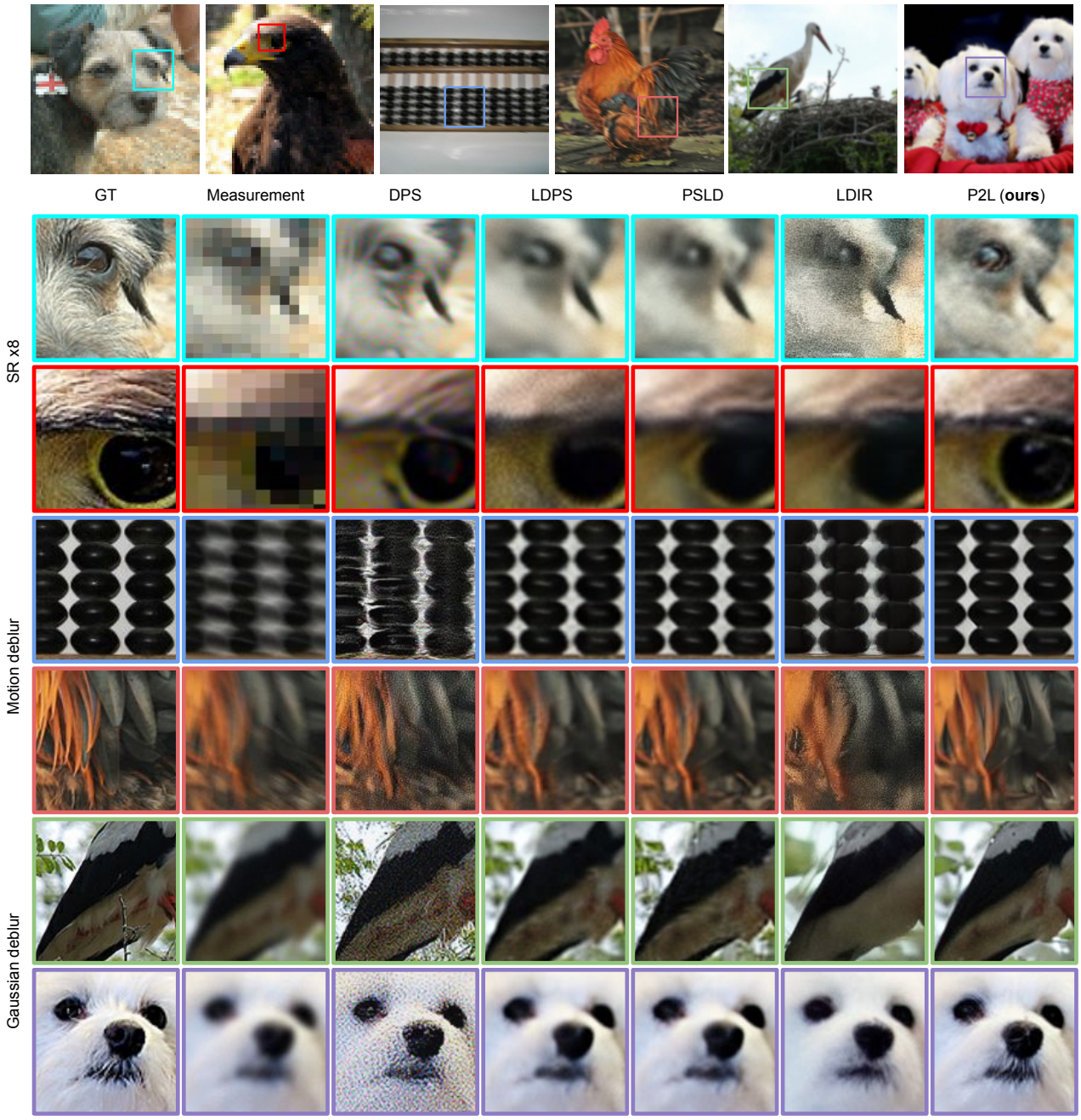


*Figure 11.* Close-up comparison on diverse inverse problem tasks. Ground truth, measurement, DPS (Chung et al., 2023b), LDPS, PSLD (Rout et al., 2023b), LDIR (He et al., 2023), and the proposed method.

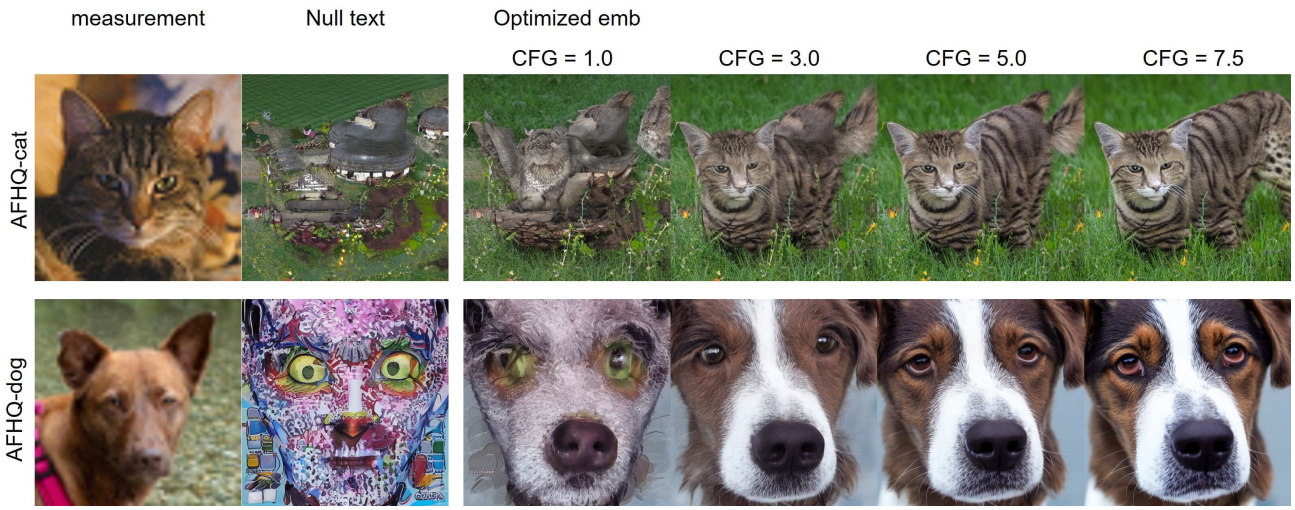


*Figure 12.* Indirect visualization of the optimized embedding through solving an inverse problem with P2L. After solving SR$\times$8 with measurements in the first column, we perform unconditional sampling by fixing the random seed, and replacing the condition with the optimized embedding by varying the CFG.

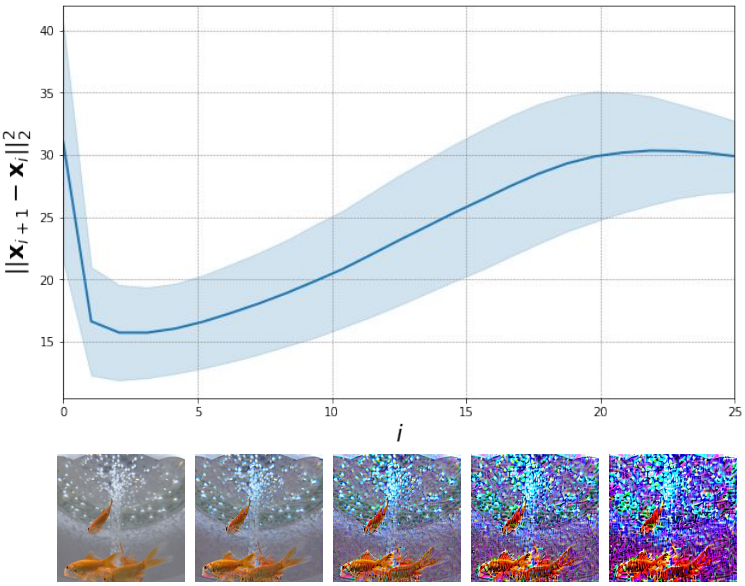


*Figure 13.* Fixed point analysis: $\mu \pm \sigma$ plotted by successive application of encoding-decoding.

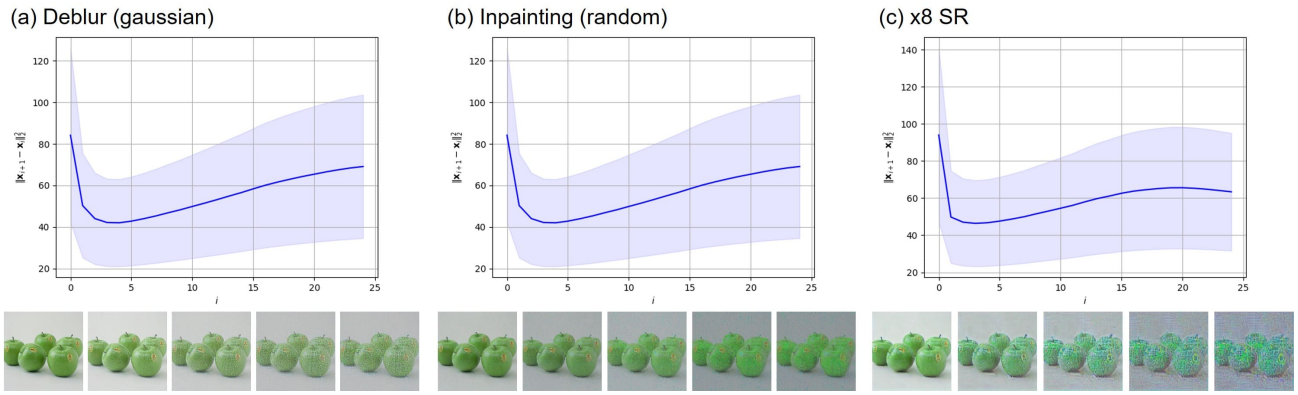


*Figure 14.* Fixed point analysis of the gluing objective in (Rout et al., 2023b) under different imaging operator $\boldsymbol{A}$.