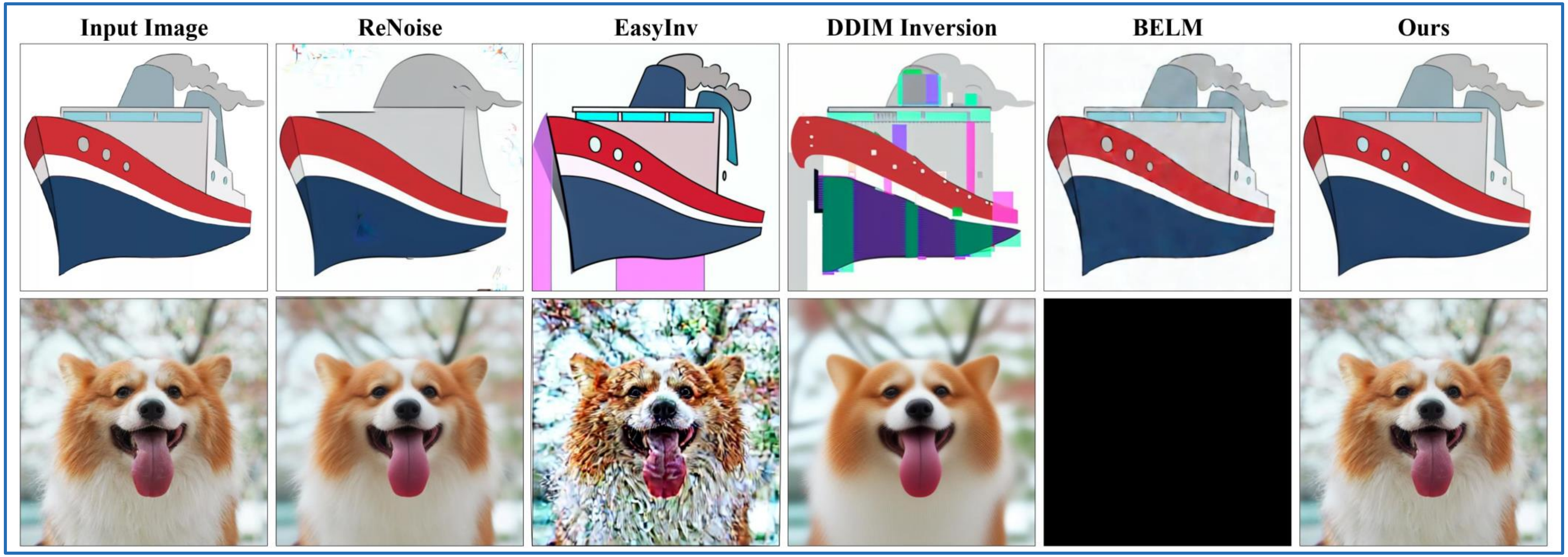
Precise Diffusion Inversion: Towards Novel Samples and Few-Step Models

Jing Zuo, Luoping Cui, Chuang Zhu, Yonggang Qi (Corresponding Author)
Beijing University of Posts and Telecommunications

NeurIPS 2025 Poster

Diffusion Inversion Problem

- Given a real image x_0, diffusion inversion aims to estimate the latent representation x_T, such that a pretrained diffusion model π_θ can reconstruct the image x_0 by applying its denoising process $\mathcal{X}$ to x_T.
- Current methods convert diffusion inversion to a fixed-point problem and solve it using numerical iterations (e.g., Anderson acceleration or Newton's method).
- However, achieving both accuracy and efficiency remains challenging, especially for few-step models (the first row) and novel samples (the second row).



Reformulating Diffusion Inversion as a Learning Problem

- We believe that gradient-based optimization is inherently more precise and efficient than numeric solvers, owing to the smooth transitions in the latent space of diffusion models. Thus, we formulate diffusion inversion as a learning problem.
- Let x_0 be an observed image, and let ϵ_T^* denote a learnable noise embedding. The objective is defined as:

$$\arg\min_{\epsilon_T^*} \|\mathcal{X}(x_T^*) - x_0\|^2,$$

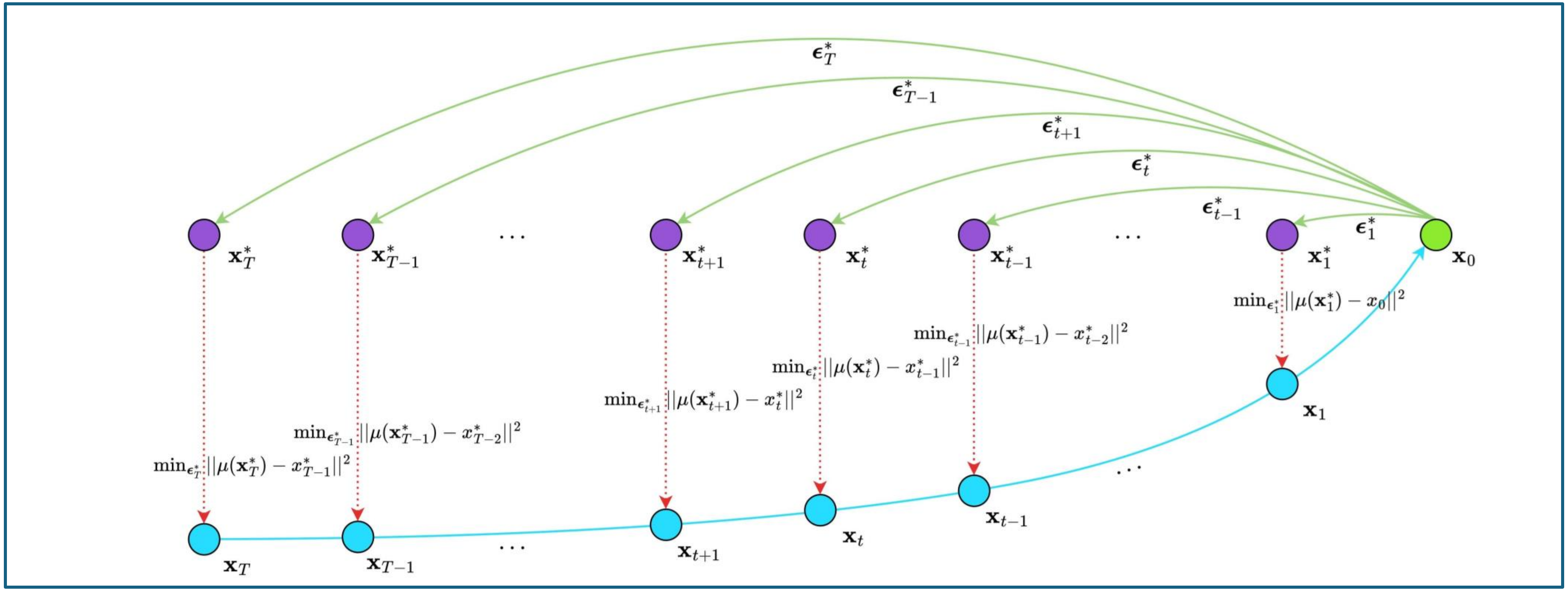
where

$$x_T^* = \sqrt{\bar{\alpha}_T} x_0 + \sqrt{1 - \bar{\alpha}_T}\, \epsilon_T^*,$$

and $\mathcal{X}: x_T \rightarrow x_{T-1} \rightarrow \cdots \rightarrow x_0$ is defined by the pretrained diffusion model.
- Directly optimizing the objective requires backpropagation through all T timesteps, which incurs prohibitive memory and computational costs.

Progressive Denoising Trajectory Learning

- To address this issue, we decompose the objective into T subproblem, in which we learn a local noise embedding ϵ_t^* enabling one denoising step reconstructs the previous latent x_{t-1}.
- By recursively solving these subproblems from $t = 1$ to T, we can obtain a precise denoising trajectory $\{x_1^*, \ldots, x_T^*\}$.



Optimization Algorithm of *PreciseInv*

Algorithm 1 *PreciseInv* for Diffusion Models with DDIM Sampler

---

Input: Real image $\mathbf{x}_0$, diffusion model $\boldsymbol{\epsilon}_\theta$, convergence threshold η, number of inference steps T
Output: Inverted latent $\mathbf{x}_T^*$
for $t = 1$ to T **do**
  Initialize $\boldsymbol{\epsilon}_t^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  if $t > 1$ **then**
    $\mathbf{x}_{t-1}^* \leftarrow \sqrt{\bar{\alpha}_{t-1}}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\,\boldsymbol{\epsilon}_{t-1}^*$
  else
    $\mathbf{x}_{t-1}^* \leftarrow \mathbf{x}_0$
  end if
  `// Apply a single DDIM Inversion step`
  $\boldsymbol{\epsilon}_t^* \leftarrow \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(\mu(\mathbf{x}_{t-1}^*) - \sqrt{\bar{\alpha}_t}\mathbf{x}_0\right)$ (μ defined in Eq. (16))
  $\mathbf{x}_t^* \leftarrow \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_t^*$
  $\mathcal{L}_{\text{rec}} \leftarrow \|\mu(\mathbf{x}_t^*) - \mathbf{x}_{t-1}^*\|^2$
  while $\mathcal{L}_{\text{rec}} < \eta$ **do**
    $\mathbf{x}_t^* \leftarrow \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_t^*$
    $\mathcal{L}_{\text{rec}} \leftarrow \|\mu(\mathbf{x}_t^*) - \mathbf{x}_{t-1}^*\|^2$
    $\boldsymbol{\epsilon}_t^* \leftarrow \boldsymbol{\epsilon}_t^* - \nabla\mathcal{L}_{\text{rec}}$
  end while
end for
return $\mathbf{x}_T^*$

Theorical Analysis

- Assumption 1. The diffusion model $\epsilon_\theta(x_t, t)$ is Lipschitz continuous in x_t with constant L_m, i.e., $\|\epsilon_\theta(x_1, t) - \epsilon_\theta(x_2, t)\| \leq L_m \|x_1 - x_2\|$.
- Assumption 2. Let $\mathcal{L}_t \coloneqq \|\mu(x_t) - \mu(x_{t-1})\|^2$. The function $\mathcal{L}_t$ has L-Lipschitz continuous gradients with respect to x_t, i.e., $\|\nabla\mathcal{L}_t(x_t) - \nabla\mathcal{L}_t(x_t')\| \leq L\|x_t - x_t'\|$.

- Theorem 1. Under Assumptions 1 and 2, the gradient descent algorithm on each subproblem $P(t) = \arg\min_{\epsilon_T^*} \|\mathcal{X}(x_t^*) - x_{t-1}^*\|^2$ with step size $\gamma < 2/L$ converges monotonically to a stationary point of $\mathcal{L}_t$.
- Theorem 2. Let $\Gamma(t) \coloneqq \|\mathcal{X}(x_t) - x_0\|^2$. Under Assumption 1 and Theorem 1, $\Gamma(T)$ admits a geometric upper bound, i.e., $\Gamma(T) \leq \delta/(1 - L_m)$.

Supported Models & Schedulers

Code is available at https://github.com/panda7777777/PreciseInv

Types	Models	Scheduler
Diffusion	SD1.4	DDIM / DDPM
	SD1.5	DDIM / DDPM
	SD2.1	DDIM / DDPM
	SDXL	DDIM / DDPM
Few-Step Models	LCM-SD1.5	DDIM / DDPM
	LCM-SDXL	DDIM / DDPM
	SDXL-Turbo	DDIM / DDPM
Rectified Flow Models	SD3	Euler

Image Reconstruction

- We further demonstrate the generalizability of **PreciseInv**.
- Moreover, the performance gains are more significant under few-step s

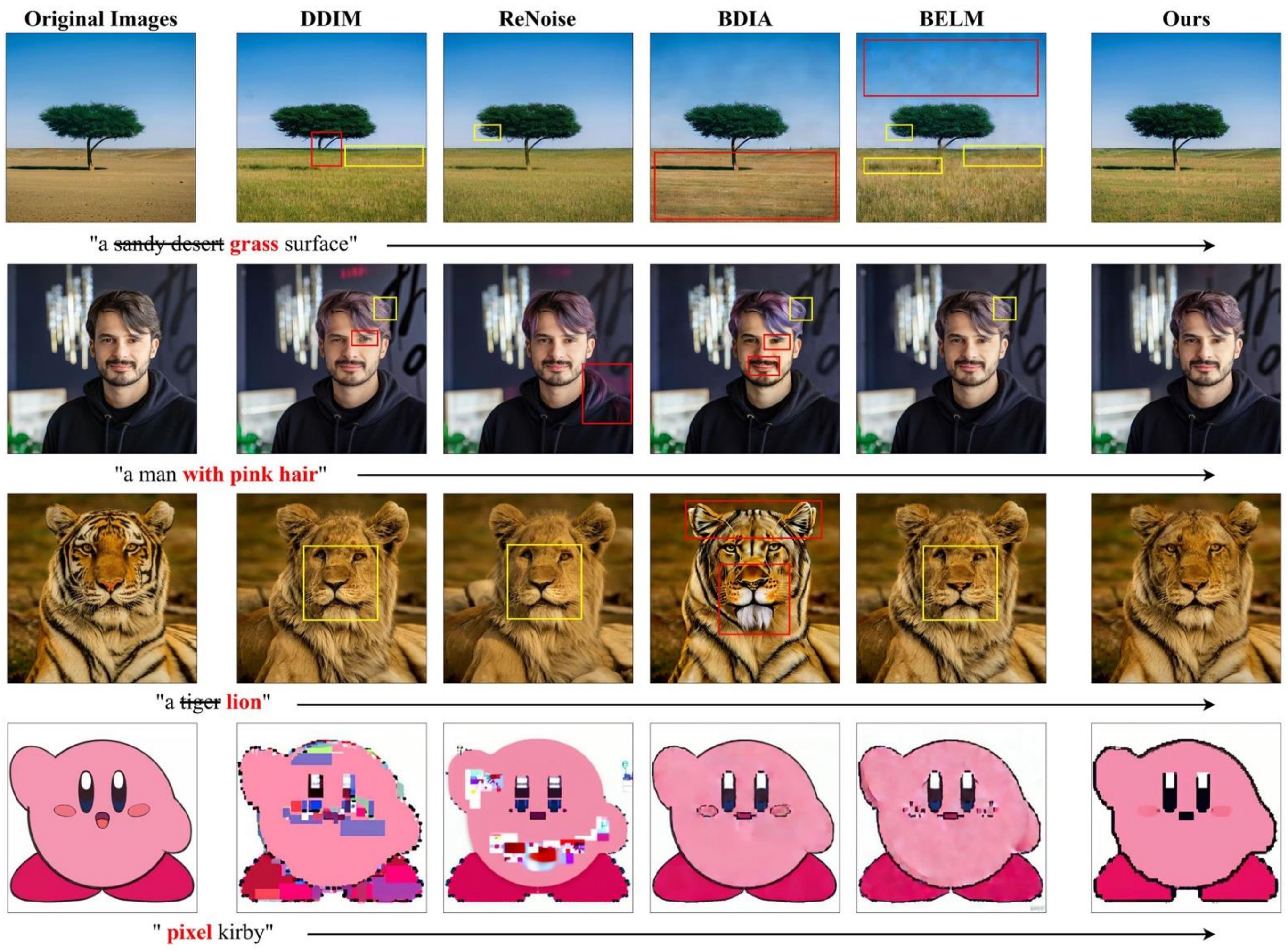
Table 2: Quantitative comparison of diffusion inversion methods for image reconstruction using LCM-SD v1.5 and SDXL on the COCO dataset.

	LCM-SD v1.5				SDXL			
	LPIPS (↓)	SSIM (↑)	PSNR (↑)	Time (s, ↓)	LPIPS (↓)	SSIM (↑)	PSNR (↑)	Time (s, ↓)
BELM	0.449	0.362	13.99	9.1	–	–	–	–
DDIM Inversion	0.626	0.420	15.74	35.5	0.492	0.464	16.24	154.8
ReNoise	0.603	0.414	15.59	83.1	0.424	0.533	18.26	27.4
EasyInv	–	–	–	–	0.194	0.683	20.89	35.1
PreciseInv ($\eta = 10^{-2}$)	0.103	0.753	25.55	**8.83**	0.185	0.763	27.00	7.43
PreciseInv ($\eta = 10^{-5}$)	**0.083**	**0.775**	**26.45**	25.54	**0.080**	**0.854**	**30.93**	41.11

Table 3: Quantitative results of image reconstruction under the few-step inference setting for different inversion methods using SD v1.4 on the COCO dataset.

	$T = 2$			$T = 4$		
	LPIPS (↓)	SSIM (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	PSNR (↑)
BELM	0.832	0.112	8.19	0.431	0.366	14.03
DDIM Inversion	0.331	0.591	20.62	0.306	0.593	20.65
ReNoise	0.199	0.671	*22.92*	0.158	0.702	23.53
EasyInv	0.563	0.357	14.97	0.537	0.332	13.66
PreciseInv	**0.077**	**0.766**	**25.93**	**0.084**	**0.762**	**25.65**

Image Editing



- We compare **PreciseInv** with recent inversion methods under the same editing pipeline to assess their editing performance.
- We demonstrate precise inversion enables more faithful and controllable prompt-based editing.

Linear Interpolation

- To probe the local behavior of PreciseInv, we interpolate between two inverted latent representations $x_T^{(1)}$ and $x_T^{(2)}$, and visualize the intermediate representations.

Thanks