DiffRGD: An Inference-Time Diffusion Guidance Through Riemannian Gradient Descent

Jia-Wei Liao[1,4], Li-Xuan Peng[2],
Mei-Heng Yueh[3], Min Sun[2], Cheng-Fu Chou[1], Jun-Cheng Chen[4]

[1]National Taiwan University,
[2]National Tsing Hua University,
[3]National Taiwan Normal University,
[4]Research Center for Information Technology Innovation, Academia Sinica
d11922016@csie.ntu.edu.tw, pullpull@citi.sinica.edu.tw

Abstract. Recently, diffusion models have been widely adopted in generative modeling and have served as foundational models for many image generation tasks. To control the generation without costly re-training or fine-tuning, many works seek inference-time guidance methods to steer the latent via a differentiable objective at inference time. However, these methods cannot effectively preserve the original Gaussian distribution because they introduce distributional drift, thereby degrading the sample quality. To address this gap, we propose DiffRGD, a distribution-aware guidance framework that explicitly preserves the latent Gaussian structure. DiffRGD formulates each sampling step as a constrained optimization problem on a spherical manifold induced by the latent Gaussian distribution, and solves it efficiently via Riemannian Gradient Descent (RGD). DiffRGD is a plug-and-play method that can be seamlessly integrated into any pre-trained diffusion model. Extensive experiments demonstrate that DiffRGD outperforms previous methods in most image restoration and conditional generation tasks. Our project page is available at https://diffrgd.github.io/.

1 Introduction

Diffusion models [14, 24, 33, 38] have achieved state-of-the-art performance in unconditional and conditional generation. Considering the prohibitively expensive training efforts of large conditional diffusion models [15, 53], many works seek inference-time guidance, where the sampler is steered by differentiable objectives without any re-training or fine-tuning. Most existing methods [4, 12, 49] inject external gradients directly into the sampling process [8]. Specifically, they formulate the guidance for each step as an optimization problem:

$$\begin{aligned} \hat{\boldsymbol{x}}_{t-1} &\sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t), \\ \boldsymbol{x}_{t-1} &= \underset{\boldsymbol{x}\in\mathcal{N}(\hat{\boldsymbol{x}}_{t-1})}{\operatorname{argmin}} \mathcal{L}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}, t-1), \boldsymbol{y}). \end{aligned} \tag{1}$$

Here $\hat{\boldsymbol{x}}_{t-1}$ denotes the noisy latent sampled from the reverse distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ parameterized by the model θ. This prediction serves as the reference point for

constructing the neighborhood constraint $\mathcal{N}(\hat{\boldsymbol{x}}_{t-1})$, which restricts the optimization process as it moves in the guidance direction induced by the guidance loss $\mathcal{L}(\cdot, \boldsymbol{y})$. Note that $\hat{\boldsymbol{x}}_0(\boldsymbol{x}, t-1)$ denotes the model's estimate of the clean sample (i.e., at timestep $t=0$ obtained from timestep $t-1$); this estimate is used to compute the guidance loss.

However, their guidance operates in the ambient space that cannot preserve the stepwise Gaussian latent distribution. This might elicit distributional drifts induced by the Jensen gap [4, 49], which ultimately degrade the clean sample quality. A representative example is DPS [4], which applies gradients from the guiding loss without any constraints. When the step size is large, the deviation can be exacerbated, whereas a small step size might lead to inadequate controllability.

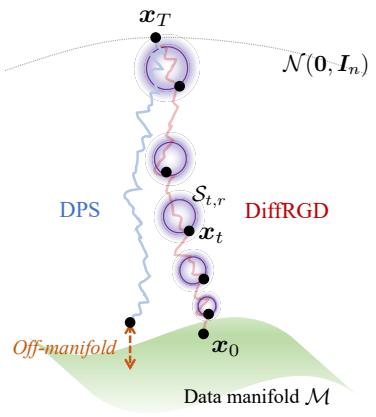

Fig. 1: Illustration of our method: Previous works perform guidance without latent-distribution-aware constraints, which can result in the off-manifold phenomena that hinder the sample quality; DiffRGD proposes a latent-distribution-aware geometry constraint tailored to the Gaussian properties of the diffusion model.

These research gaps motivate us to ask: *Can we design latent-distribution-aware guidance that ensures the guided sample lies within the original latent distribution?*

In this paper, we propose *DiffRGD*, an inference-time diffusion guidance framework based on *Riemannian optimization* [1]. We follow the problem formulation of previous works defined in Equation 1 with our proposed novel geometry constraints. Our key insight is that since each DDIM sampling timestep [34] defines an *isotropic Gaussian* of latent distribution, we can construct a suitable constraint that respects the Gaussian properties to perform Riemannian Gradient Descent (RGD). DiffRGD constructs the constraint as a spherical manifold induced from the latent distribution via our proposed polar decomposition property. We also prove that our method converges to a stationary point. In addition, akin to previous methods, DiffRGD is a plug-and-play method and can be seamlessly applied to any pre-trained diffusion model.

We conduct extensive experiments, including synthetic data and real-world tasks. We create a 2D synthetic dataset to demonstrate our distribution-preserving superiority over DPS, please refer to the Appendix. In real-world tasks, DiffRGD outperforms previous methods in most image restoration [4, 5, 12, 20, 26, 32, 35, 44, 49, 55, 56] (e.g., inpainting, super-resolution, deblurring, denoising) and conditional generation tasks [2, 12, 36, 49, 51, 55] (e.g., segmentation map, sketch, FaceID, style). Additionally, we reimplement all the compared methods under the Diffusers framework [30].

Our main contributions are summarized as follows:

1. We identify a polar-decomposition property that reveals a distribution-induced geometry for diffusion guidance. Leveraging this insight, we propose a Riemannian guidance framework that respects the latent distribution geometry and mitigates sample degradation caused by off-manifold distribution drift.
2. We provide an analysis of distribution preservation to validate the effectiveness of the proposed method and prove the convergence of the algorithm. In practice, the proposed geometry-aware optimization converges faster than ADMM-based methods.
3. We demonstrate the effectiveness of DiffRGD across four image restoration tasks and three conditional generation tasks, achieving state-of-the-art performance without re-training or fine-tuning.

2 Related Work

2.1 Diffusion Models for Conditional Generation

Diffusion models have emerged as a dominant class of generative models due to their strong modeling capacity and sample quality [14, 18, 33, 34, 38]. Recent advancements have enabled scalable and foundational implementations, such as Stable Diffusion [10, 31] and FLUX [22, 23], which have been widely adopted in commercial applications. To enable controllable generation, several guidance strategies have been introduced to steer the sampling process.

Classifier guidance (CG) [8] utilizes an external classifier to steer the sampling process, while classifier-free guidance (CFG) [15] achieves guidance by interpolating conditional and unconditional model outputs without requiring an external classifier. These techniques have been widely adopted across diverse downstream applications, including text-to-image synthesis [31], conditional image generation [2, 25, 28, 36, 42, 46, 51, 53], and test-time adaptation [41].

2.2 Inference-Time Diffusion Guidance

Despite the powerful capability of diffusion models, their training demands substantial computational resources and large-scale datasets. Fine-tuning or retraining diffusion models for every new downstream task or modality is prohibitively expensive and leads to high maintenance overhead. To address this, several works have proposed inference-time guidance methods that refine the latent during inference without modifying the original model parameters.

DPS. DPS [4] derives the guidance by maximizing posterior likelihood from the inverse problem, and FreeDoM [51] derives the guidance by minimizing an energy function that measures the conditional alignment. They both share a similar update rule that does not constrain the updates to respect the latent distribution. Although they are intuitive and effective, they often suffer from off-manifold updates if the step size is too large or insufficient guidance when the step size is too small, both of which can degrade the final clean sample quality.

MPGD. To alleviate the off-manifold issue, MPGD [12] proposes a Projected Gradient Descent (PGD) framework with a linear manifold hypothesis. At each timestep, it estimates the clean data $\boldsymbol{x}_0$, projects the guiding gradient onto the tangent space of the data manifold using a VAE encoder, and updates the latent accordingly. While this projection mitigates deviation from the data manifold, it assumes local linearity and introduces instability due to the reliance on a perfect encoder, which in practice is infeasible. This limits its general applicability.

DSG. Alternatively, DSG [49] leverages a spherical Gaussian structure and projects the guided latent to a fixed radius hypersphere. However, their proposed hypersphere cannot fully adhere to the latent distribution's Gaussian properties, even degrading to a uniform distribution. In contrast, we propose a polar decomposition directly from the latent and a radius sampling technique to maintain the Gaussian properties for the optimization constraints.

ADMMDiff. Another line of work, such as ADMMDiff [55], formulates diffusion guidance as a constrained optimization problem and incorporates the Alternating Direction Method of Multipliers (ADMM) [3, 11, 43] into the sampling process. While theoretically grounded, it utilizes the proximal operator and inner-loop iteration, leading to slow convergence. It also requires careful tuning of multiple hyperparameters, making it less flexible in various applications.

We seek to address the research gap by proposing a more suitable geometric constraint that is directly induced from the original latent distribution and solving it via Riemannian Gradient Descent [1], which provides efficient computation with a convergence guarantee.

3 Method

Denoising Diffusion Probabilistic Models (DDPMs) [14] generate data by learning to reverse a forward noising process, in another equivalent view, is learning to perform sampling across different levels of noise-perturbed data distribution [37, 38]. DDIM [34] generalized the DDPM formulation with parameters σ_t controlling the noise level during sampling to enable sample acceleration and a unified formulation. Here, we follow their formulation to explain our motivation. Starting from a noisy latent $\boldsymbol{x}_t$ at timestep t, DDIM defines the next-step latent $\boldsymbol{x}_{t-1}$ during sampling, i.e., denoising, as sampling from a Gaussian distribution parameterized by the model prediction.

$$\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}_n),$$

where the mean is defined by

$$\boldsymbol{\mu}_t := \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t) + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t),$$

with given DDIM sampling schedule parameters $\{\bar{\alpha}_t\}_{t=1}^T$ and the $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t)$ is the predicted clean sample, estimated by Tweedie's formula [9]:

$$\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t) := \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right),$$

in which the $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ denotes the predicted noise from a diffusion model.

In the unconditional sampling setting, the latents across sampling timesteps will form an isotropic Gaussian coupled with the timestep t. However, when applying inference-time conditional sampling, the guidance often deviates the latent from the original Gaussian distribution. As the sampling step proceeds, the deviation will accumulate, making the final clean sample deviate from the data manifold, resulting in lower quality. To address the off-manifold problem, we propose DiffRGD, an inference-time guidance method based on Riemannian Gradient Descent (RGD) [1]. Our key insight is that for each sampling timestep, the latent follows a time-dependent Gaussian distribution, and by the properties of an isotropic Gaussian, we can construct a distribution-induced spherical manifold as a proxy to perform RGD while ensuring the optimized latent still lies within the original Gaussian distribution. The list of notations used in this section is provided in the Appendix.

3.1 Polar Decomposition of Isotropic Gaussian

In this section, we characterize the geometric structure induced by the latent distribution at each diffusion timestep. We begin by showing that the isotropic Gaussian latent admits a polar decomposition into statistically independent radial and directional components, as stated in the following proposition.

Proposition 1 (Polar Decomposition of Isotropic Gaussian). *At each diffusion timestep t, the latent $\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}_n)$ follows an isotropic Gaussian distribution centered at $\boldsymbol{\mu}_t$. It admits a polar decomposition:*

$$\boldsymbol{x}_t = \boldsymbol{\mu}_t + \sigma_t r \boldsymbol{u},$$

where $r \sim \chi(n)$ is a chi-distributed random variable with n degrees of freedom, and $\boldsymbol{u} \sim \mathrm{Unif}(\mathbb{S}^{n-1})$ is a unit vector uniformly distributed on the sphere. r and $\boldsymbol{u}$ are statistically independent.

The insight of Proposition 1 is that we aim to construct the spherical manifold based on the same density as the original Gaussian of $\boldsymbol{x}_t$. The radius $\sigma_t r$ follows a scaled-chi distribution centered close to $\sqrt{n}\sigma_t$ with two-sided probability decreasing. We show the conceptual illustration in Figure 2. We also provide an asymptotic analysis for our scaled-chi distribution behavior when n is large in the Appendix.

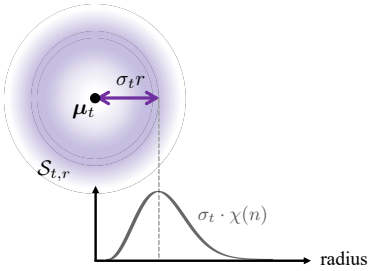

Fig. 2: Gaussian distribution-aware spherical manifold construction.

This polar decomposition decouples the sampled r and direction $\boldsymbol{u}$. Here, we propose to fix the sampled r to construct the spherical manifold as the feasible

Algorithm 1 DiffRGD

Input: Reference image $\boldsymbol{y} \in \mathbb{R}^n$, Diffusion model $\boldsymbol{\epsilon}_\theta$, loss function $\mathcal{L}$, DDIM sampling schedule parameters $\{\bar{\alpha}_t\}_{t=1}^T$, noise level $\sigma_t > 0$, guidance strengths $\{\eta_t^{(k)}\}_{t=1}^T$, guidance inner iteration $K \in \mathbb{N}$.

$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$.
for $t = T$ to 1 **do**
  $r \sim \chi(n)$.
  $\boldsymbol{u} \sim \mathrm{Unif}(\mathbb{S}^{n-1})$.
  $\boldsymbol{\mu}_t \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t) + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$.
  $\boldsymbol{x}_{t-1}^{(0)} \leftarrow \boldsymbol{\mu}_t + \sigma_t r \boldsymbol{u}$.
  for $k = 0$ to $K - 1$ **do**
    $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{x}_{t-1}^{(k)}} \mathcal{L}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_{t-1}^{(k)}, t-1), \boldsymbol{y})$. ▷ Guidance gradient
    $\mathrm{grad}_{\mathcal{S}_{t,r}} \mathcal{L} \leftarrow \Pi_{\mathrm{T}_{\boldsymbol{x}_{t-1}^{(k)}}\mathcal{S}_{t,r}}(\boldsymbol{g})$. ▷ Riemannian gradient
    $\boldsymbol{x}_{t-1}^{(k+1)} \leftarrow \mathrm{R}_{\boldsymbol{x}_{t-1}^{(k)}}(-\eta_t^{(k)}\,\mathrm{grad}_{\mathcal{S}_{t,r}}\mathcal{L})$. ▷ Retraction
  end for
  $\boldsymbol{x}_{t-1} \leftarrow \boldsymbol{x}_{t-1}^{(K)}$.
end for

Return: Guided sample $\boldsymbol{x}_0$.

set for RGD. The goal is to find a suitable direction $\boldsymbol{u}$ toward the guidance. Specifically, the spherical manifold is defined as:

$$\mathcal{S}_{t,r} := \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{\mu}_t\|_2 = \sigma_t r\}\,. \tag{2}$$

In contrast to previous unconstrained gradient updates [4,51], our constraint could preserve the distribution-induced geometry of the diffusion process while allowing the directional refinement toward the guidance direction.

Problem Formulation. We follow previous works [12] to formulate the per-timestep guidance as a constrained optimization problem. Given a conditional input $\boldsymbol{y}$, we formulate the latent refinement at each timestep as:

$$\boldsymbol{x}_{t-1} = \underset{\boldsymbol{x} \in \mathcal{S}_{t,r}}{\mathrm{argmin}}\, \mathcal{L}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}, t-1), \boldsymbol{y}). \tag{3}$$

Since we constrain the optimization respecting the feasible set defined in Equation 2, this optimization naturally respects the Gaussian properties of the diffusion process while improving the sample alignment toward the condition $\boldsymbol{y}$.

3.2 Riemannian Gradient Descent for Diffusion Guidance

To solve the constrained optimization problem in Equation 3, we employ Riemannian Gradient Descent (RGD) on the spherical manifold $\mathcal{S}_{t,r}$. We first define the geometric components for the optimization step as follows:

Tangent Space. The tangent space of the spherical manifold $\mathcal{S}_{t,r}$ at a point $\boldsymbol{x} \in \mathcal{S}_{t,r}$ consists of all directions orthogonal to the radial vector:

$$\mathrm{T}_{\boldsymbol{x}}\,\mathcal{S}_{t,r} := \left\{ \boldsymbol{v} \in \mathbb{R}^n \mid (\boldsymbol{x} - \boldsymbol{\mu}_t)^\top \boldsymbol{v} = 0 \right\}.$$

Riemannian Metric. We endow the manifold $\mathcal{S}_{t,r}$ with the standard Riemannian metric induced by the Euclidean inner product. For any two vectors in the tangent space $\boldsymbol{v}, \boldsymbol{w} \in \mathrm{T}_{\boldsymbol{x}}\,\mathcal{S}_{t,r}$, the inner product is defined as:

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle_{\boldsymbol{x}} := \boldsymbol{v}^\top \boldsymbol{w}.$$

Projection Operator. The projection of a vector $\boldsymbol{v} \in \mathbb{R}^n$ onto the tangent space $\mathrm{T}_{\boldsymbol{x}}\,\mathcal{S}_{t,r}$ is given by:

$$\Pi_{\mathrm{T}_{\boldsymbol{x}}\,\mathcal{S}_{t,r}}(\boldsymbol{v}) := \left(\boldsymbol{I}_n - \frac{(\boldsymbol{x} - \boldsymbol{\mu}_t)(\boldsymbol{x} - \boldsymbol{\mu}_t)^\top}{\|\boldsymbol{x} - \boldsymbol{\mu}_t\|_2^2} \right) \boldsymbol{v},$$

This projection removes the radial part of $\boldsymbol{v}$ to ensure the update direction remains tangent to the spherical manifold.

Retraction. After taking a step along the tangent space, the resulting latent may leave the spherical manifold. We apply a retraction operation to map the updated latent back to the spherical manifold by:

$$\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{v}) := \boldsymbol{\mu}_t + \sigma_t r \cdot \frac{\boldsymbol{x} - \boldsymbol{\mu}_t + \boldsymbol{v}}{\|\boldsymbol{x} - \boldsymbol{\mu}_t + \boldsymbol{v}\|_2},$$

which ensures the latent updates lying on the same spherical manifold specified by the radius $\sigma_t r$.

Fig. 3 shows the guidance step in DiffRGD. For each diffusion timestep t, given the predicted mean $\boldsymbol{\mu}_t$ by DDIM and the radius $\sigma_t r$, the latent $\boldsymbol{x}_{t-1}^{(0)}$ lies on the constructed spherical manifold $\mathcal{S}_{t,r}$ before applying guidance. Starting from this initial latent, we perform K Riemannian optimization steps to refine the latent by the guiding loss while ensuring it lies on the constructed manifold.

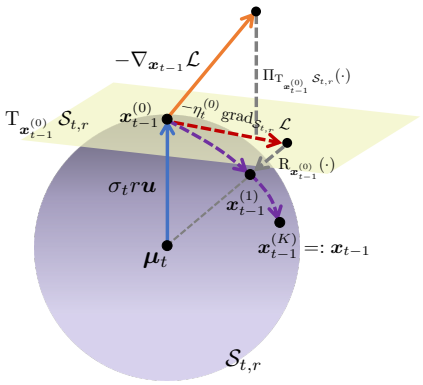

Fig. 3: An illustration of DiffRGD.

In each Riemannian optimization iteration, we compute the negative Euclidean gradient $-\nabla_{\boldsymbol{x}_{t-1}^{(k)}} \mathcal{L}$, where $\mathcal{L}(\cdot, \boldsymbol{y})$ denotes the guidance objective conditioned on $\boldsymbol{y}$. Then, we use this gradient to obtain the Riemannian gradient and perform retraction for the guidance update. Specifically, to respect the spherical constraint,

this gradient is first projected onto the tangent space $\mathrm{T}_{\boldsymbol{x}_{t-1}^{(k)}} \mathcal{S}_{t,r}$, yielding the Riemannian gradient:

$$\mathrm{grad}_{\mathcal{S}_{t,r}} \mathcal{L} := \Pi_{\mathrm{T}_{\boldsymbol{x}_{t-1}^{(k)}} \mathcal{S}_{t,r}} \left(\nabla_{\boldsymbol{x}_{t-1}^{(k)}} \mathcal{L}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_{t-1}^{(k)}, t-1), \boldsymbol{y}) \right).$$

We then update the latent along the negative Riemannian gradient direction with guidance strength $\eta_t^{(k)} > 0$ via a retraction-based step:

$$\boldsymbol{x}_{t-1}^{(k+1)} = \mathrm{R}_{\boldsymbol{x}_{t-1}^{(k)}} \left(-\eta_t^{(k)} \, \mathrm{grad}_{\mathcal{S}_{t,r}} \mathcal{L} \right).$$

The retraction ensures that each update remains on the spherical manifold $\mathcal{S}_{t,r}$, preventing the radial deviation that naive Euclidean gradient updates would otherwise introduce. After K iterations, we obtain the optimized latent $\boldsymbol{x}_{t-1}^{(K)}$, which serves as the final guided sample $\boldsymbol{x}_{t-1}$.

The full algorithm is shown in Algorithm 1. This framework ensures each optimization step remains on the spherical manifold induced by the isotropic Gaussian, thereby preserving the validity of the sampling process and mitigating quality degradation caused by off-manifold phenomena. A detailed analysis of the distribution preservation property is provided in the Appendix.

3.3 Convergence Analysis

To ensure the stability and reliability of our guidance framework, it's essential to study the convergence behavior of the optimization steps. In this section, we theoretically justify the convergence of our optimization loop. In Theorem 1, we show that at each diffusion sampling step, our method asymptotically converges to a stationary point.

Theorem 1 (Convergence to Stationary Point of DiffRGD). *Let $\boldsymbol{f}_t : \mathcal{S}_{t,r} \to \mathbb{R}$ be the guidance objective at timestep t, defined as $\boldsymbol{f}_t(\boldsymbol{x}) := \mathcal{L}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}, t-1), \boldsymbol{y})$, and let $\{\boldsymbol{x}_{t-1}^{(k)}\}$ denote the sequence produced by Algorithm 1 at timestep t. Assume that $\boldsymbol{f}_t$ is L_t-smooth on the spherical manifold $\mathcal{S}_{t,r}$, and the guidance strength $0 < \eta_{\min} \leq \eta_t^{(k)} \leq \frac{1}{L_t}$. Then the norm of the Riemannian gradient satisfies*

$$\left\| \mathrm{grad}_{\mathcal{S}_{t,r}} \boldsymbol{f}_t(\boldsymbol{x}_{t-1}^{(k)}) \right\|_2 \to 0 \text{ as } k \to \infty.$$

with a convergence rate $\mathcal{O}(\frac{1}{\sqrt{k}})$.

This theorem shows that, under the assumption of L-smoothness, Algorithm 1 converges to a stationary point at each timestep with a sublinear rate. In our task-specific application, we observe that the Riemannian gradient norm typically becomes small within three iterations. Therefore, in practice, we fix the number of optimization steps to $K = 3$, and provide the convergence curves of the gradient norm across several timesteps in the Appendix.

4 Experiment Results

For a fair comparison, we reimplement several diffusion guidance baselines under identical data and preprocessing, including DPS [4], FreeDoM [51], MPGD [12], DSG [49], and ADMMDiff [55]. Detailed algorithmic descriptions of each method, along with the full set of hyperparameters used in our experiments, are provided in the Appendix. We follow the official configurations reported in the original papers wherever available. For methods lacking complete parameter specifications in either the paper or public codebase, we report the best-performing hyperparameter settings. All experiments used the DDIM sampling schedule with $\sigma_t = \sqrt{(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)} \cdot \sqrt{1-\bar{\alpha}_t/\bar{\alpha}_{t-1}}$ and were implemented in PyTorch [29] with Hugging Face Diffusers [30] and ran on a single NVIDIA GeForce RTX 4090 GPU. In the following sections, we evaluate baseline methods and our proposed approach on image restoration tasks (Section 4.1) and conditional generation tasks (Section 4.3).

4.1 Image Restoration

We evaluate our proposed method alongside representative baseline approaches on four image restoration tasks. The objective in each task is to reconstruct a clean image $\boldsymbol{x}$ from a degraded observation $\boldsymbol{y}$. Each task is formulated as a noisy linear inverse problem [4,20,21,40,44,48,52,56] modeled by the forward process:

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n},\ \boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, 0.05^2\boldsymbol{I})$$

where $\mathcal{A}(\cdot)$ denotes a task-specific linear degradation operator. Given the noisy measurement $\boldsymbol{y}$, the goal is to reconstruct $\hat{\boldsymbol{x}}_0$ conditioned on a diffusion prior such that $\mathcal{A}(\hat{\boldsymbol{x}}_0)$ remains consistent with $\boldsymbol{y}$. To enable conditional generation under each inverse setting, we define the guidance objective using an ℓ_2 loss:

$$\mathcal{L}_{\text{inv}}(\hat{\boldsymbol{x}}_0, \boldsymbol{y}) = \|\mathcal{A}(\hat{\boldsymbol{x}}_0) - \boldsymbol{y}\|_2.$$

Implementation Details. We consider four inverse problem tasks using 150 images from FFHQ 256 × 256 [19] validation datasets. We utilize pre-trained diffusion models provided by [4,8] as the priors for all guidance frameworks. Each inverse problem is defined by a different degradation operator $\mathcal{A}$: (i) **Inpainting:** A random binary mask is applied to simulate missing pixels, where each pixel has a 70% chance of being masked. (ii) **Super-Resolution:** Input images are downsampled by a factor of 4 using bicubic interpolation. (iii) **Gaussian Deblurring:** A Gaussian blur kernel of size 31 × 31 with a standard deviation 3.0 is applied to simulate blur degradation. (iv) **Motion Deblurring:** A motion blur kernel of size 61 × 61 is applied with an intensity of 0.5, and the direction of motion is randomly sampled.

Table 1: Quantitative results using DDIM 1,000 sampling steps on 150 samples from the FFHQ 256 × 256 validation set for image restoration tasks. Metrics are averaged over 100 bootstrap runs. * indicates a statistically significant improvement ($p < 0.05$) over the second-best method.

Task	Method	Venue	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Inpainting	DPS [4]	ICLR 2023	30.44	0.863	0.153	42.68
	MPGD [12]	ICLR 2024	27.51	0.724	0.256	68.24
	DSG [49]	ICML 2024	31.03	0.866	0.144	36.30
	ADMMDiff [55]	CVPR 2025	<u>32.38</u>	<u>0.899</u>	<u>0.119</u>	<u>29.52</u>
	DiffRGD (Ours)	ECCV 2026	**34.04***	**0.926***	**0.096***	**21.88***
Super-Resolution 4×	DPS [4]	ICLR 2023	26.03	0.727	0.260	80.05
	MPGD [12]	ICLR 2024	24.40	0.614	0.354	101.50
	DSG [49]	ICML 2024	<u>26.71</u>	<u>0.737</u>	<u>0.256</u>	<u>74.67</u>
	ADMMDiff [55]	CVPR 2025	26.48	0.712	0.297	96.69
	DiffRGD (Ours)	ECCV 2026	**27.77***	**0.783***	**0.220***	**63.94***
Gaussian Deblurring	DPS [4]	ICLR 2023	25.88	0.721	0.237	69.38
	MPGD [12]	ICLR 2024	24.07	0.576	0.328	95.12
	DSG [49]	ICML 2024	**27.45**	0.751	0.259	<u>75.85</u>
	ADMMDiff [55]	CVPR 2025	26.57	**0.757**	<u>0.226</u>	79.30
	DiffRGD (Ours)	ECCV 2026	<u>26.80</u>	**0.757**	**0.218***	**63.83***
Motion Deblurring	DPS [4]	ICLR 2023	24.47	0.685	0.271	80.75
	MPGD [12]	ICLR 2024	23.15	0.569	0.357	106.99
	DSG [49]	ICML 2024	<u>26.80</u>	0.709	0.290	87.96
	ADMMDiff [55]	CVPR 2025	**27.26**	**0.778**	**0.222**	72.92
	DiffRGD (Ours)	ECCV 2026	25.84	<u>0.736</u>	<u>0.250</u>	**72.26**

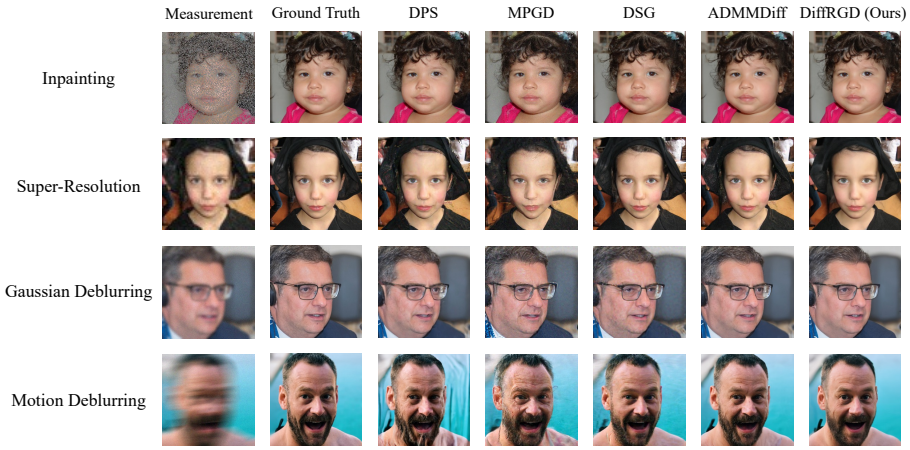

Fig. 4: Qualitative comparison for image restoration tasks on FFHQ 256 × 256 validation set. Our proposed method achieves better restoration results with fewer artifacts than other compared methods.

Table 2: Quantitative results using DDIM 100 sampling steps on 150 samples from the FFHQ 256 × 256 validation set for deblurring tasks. Metrics are averaged over 100 bootstrap runs. * indicates a statistically significant improvement ($p < 0.05$) over the second-best method.

Method	Gaussian Deblurring			Motion Deblurring		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DPS [4]	24.28	0.680	0.302	20.63	0.580	0.372
MPGD [12]	24.60	0.610	0.312	23.13	0.579	0.356
DSG [49]	<u>27.02</u>	<u>0.758</u>	<u>0.261</u>	<u>24.81</u>	<u>0.705</u>	**0.290**
ADMMDiff [55]	26.01	0.734	0.278	23.16	0.664	0.327
DiffRGD (Ours)	**27.46***	**0.781***	**0.235***	**24.88**	**0.717**	<u>0.294</u>

Table 3: Quantitative results using DDIM 100 sampling steps on 150 samples from the ImageNet 256 × 256 validation set for inpainting and super-resolution. Metrics are averaged over 100 bootstrap runs. * indicates a statistically significant improvement ($p < 0.05$) over the second-best method.

Method	Inpainting			Super-Resolution 4×		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DPS [4]	23.24	0.633	0.349	20.26	0.500	0.449
MPGD [12]	26.15	0.711	0.323	21.50	0.531	0.478
DSG [49]	26.58	<u>0.754</u>	<u>0.255</u>	22.82	0.585	<u>0.415</u>
ADMMDiff [55]	<u>27.97</u>	0.744	0.303	<u>24.45</u>	<u>0.633</u>	0.430
DiffRGD (Ours)	**31.16***	**0.875***	**0.208***	**24.95***	**0.663***	**0.379***

Evaluation Protocol. We evaluate performance using PSNR [16], SSIM [45], LPIPS [54], and FID [13] (details are provided in the Appendix). To increase the robustness of the metrics and assess statistical significance, we perform bootstrap resampling [39]. For each method, we report the mean performance across all metrics over 100 bootstrap runs, where 100 samples are randomly drawn from the 150-sample evaluation set in each run.

Quantitative Results. Table 1 reports quantitative results using 1,000 DDIM sampling steps, where our method outperforms other baselines on inpainting, super-resolution, and Gaussian deblurring. To assess the robustness of our method with fewer denoising timesteps, we conduct experiments using 100 steps and report in Table 2. As the number of timesteps decreases, DPS and ADMMDiff exhibit noticeable performance degradation, whereas DiffRGD maintains strong performance. We further evaluate inpainting and super-resolution on ImageNet 256 × 256 [6] (Table 3), where DiffRGD continues to outperform the baselines. Additional quantitative results are provided in the Appendix.

Qualitative Results. Figure 4 shows the qualitative results across four image restoration tasks. We can observe that DPS and MPGD introduce some perceptible artifacts in the background across four tasks; DSG performs well on

inpainting and super-resolution tasks, but introduces some artifacts in both deblurring tasks. DiffRGD achieves comparable visual quality with ADMMDiff, but with slightly better artifact control and demands less time.

4.2 Super-Resolution under Various Subsampling Rates

We further evaluate our method using the Stable Diffusion model [31] under various subsampling rates. Specifically, the original FFHQ images at a resolution of 1024×1024 are downsampled to 64×64 and then super-resolved to 512×512 ($8\times$) and 768×768 ($12\times$). As reported in Table 4, DiffRGD consistently outperforms both DPS and DSG across both upsampling scales in terms of quantitative metrics. Qualitative results are presented in Figure 5, which shows that higher resolutions yield sharper image details. However, the $12\times$ outputs show greater discrepancy in fine-grained structures compared to $8\times$, as reflected in Table 4: while the $12\times$ results achieve higher SSIM, they exhibit lower PSNR.

Fig. 5: Qualitative results of super-resolution by our DiffRGD method with Stable Diffusion under $\times 8$ and $\times 12$ subsampling rates.

Table 4: Quantitative results of $8\times$ and $12\times$ super-resolution with Stable Diffusion.

Method	$\times 8$ ($64 \to 512$)		$\times 12$ ($64 \to 768$)	
	PSNR ↑	**SSIM** ↑	**PSNR** ↑	**SSIM** ↑
DPS [4]	25.67	0.689	24.99	0.689
DSG [49]	25.38	0.679	26.02	**0.709**
DiffRGD (Ours)	**26.41**	**0.700**	**26.39**	**0.709**

Table 5: Quantitative results using DDIM 100 sampling steps on 150 samples from the CelebA-HQ 256 × 256 validation set for conditional image generation tasks. Metrics are averaged over 100 bootstrap runs. * indicates a statistically significant improvement ($p < 0.05$) over the second-best method.

Method	Segmentation Map			Sketch			FaceID		
	mIoU ↑	FID ↓	KID ↓	Sketch-ℓ_2 ↓	FID ↓	KID ↓	FaceID-ℓ_2 ↓	FID ↓	KID ↓
FreeDoM [51]	0.622	156.02	0.057	30.85	101.90	0.017	0.557	127.05	0.032
DSG [49]	0.750	117.48	<u>0.028</u>	<u>21.36</u>	107.00	0.024	<u>0.340</u>	<u>95.27</u>	<u>0.014</u>
ADMMDiff [55]	<u>0.758</u>	<u>101.86</u>	0.031	30.82	<u>97.52</u>	<u>0.014</u>	0.346	100.81	<u>0.014</u>
DiffRGD (Ours)	**0.804***	**96.10***	**0.026**	**19.48***	**87.82***	**0.012**	**0.303***	**93.80**	**0.011**

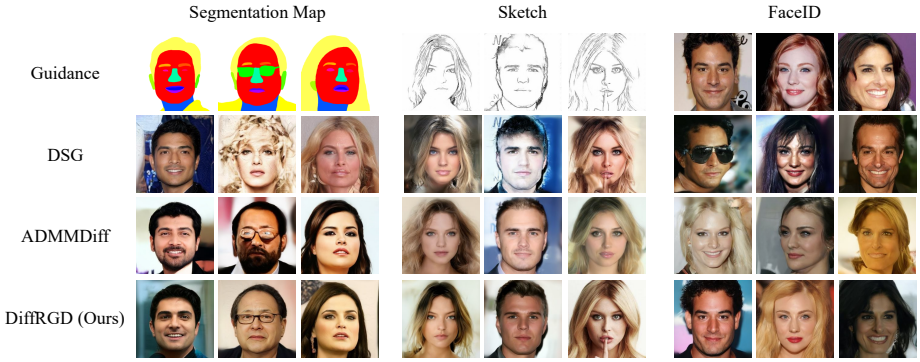

Fig. 6: Qualitative results for conditional generation tasks on CelebA-HQ 256 × 256 validation set. (a) segmentation maps (b) sketches (c) FaceID. Our method generates results that better align with the given conditions than other state-of-the-art methods.

4.3 Conditional Generation

We further evaluate our method against representative baselines on three conditional generation tasks: segmentation maps, sketches, and FaceID guidance for human face generation. We also provide style-guided generation results in the Appendix.

Given a conditional image $\boldsymbol{y}$, we extract task-specific features using a pre-trained model ψ_θ. The conditional guidance loss is then defined as:

$$\mathcal{L}_{\text{cond}}(\hat{\boldsymbol{x}}_0, \boldsymbol{y}) = \|\psi_\theta(\hat{\boldsymbol{x}}_0) - \psi_\theta(\boldsymbol{y})\|_2.$$

Implementation Details. We adopt pre-trained diffusion models provided by [27] as the generative backbone. The three tasks are defined as follows: (i) **Segmentation Map Guidance:** We use a pre-trained BiSeNet [50] to extract face parsing maps from both generated and reference images. (ii) **Sketch Guidance:** A pre-trained AODA model [47] is used to parse structural cues from input sketches. (iii) **FaceID Guidance:** We utilize a pre-trained ArcFace model [7] to extract identity embeddings for evaluating face similarity.

Evaluation Protocol. We evaluate our method on three conditional generation tasks using 150 validation images from the CelebA-HQ 256×256 [17] dataset. To increase the robustness of the metrics, we perform bootstrap resampling [39]. For each method, we report the mean performance across all metrics over 100 bootstrap runs. Details of the evaluation metrics are provided in the Appendix.

Quantitative Results. Table 5 presents the quantitative results, showing that our method consistently outperforms existing state-of-the-art inference-time methods. The generated samples exhibit improved visual fidelity and demonstrate stronger alignments with the given conditional inputs.

Qualitative Results. Figure 6 presents the qualitative results. DSG shows unstable control and often produces noticeable artifacts. DiffRGD and ADMMDiff generate samples that align well with the guidance, while DiffRGD yields more natural and visually appealing results under the same time constraint.

4.4 Influence of Guidance Strength

We investigate the influence of guidance strength η_t on DSG and DiffRGD. Figure 7 shows qualitative results under segmentation map guidance. DiffRGD better aligns with the conditioning signals while preserving natural visual quality across a wide range of guidance strengths. In contrast, DSG shows poor controllability at small guidance strengths and severe degradation at large guidance strengths. Table 6 reports quantitative results on 100 CelebA-HQ 256×256 samples, further showing that DiffRGD achieves a better mIoU–FID trade-off than DSG and remains stable across guidance strengths.

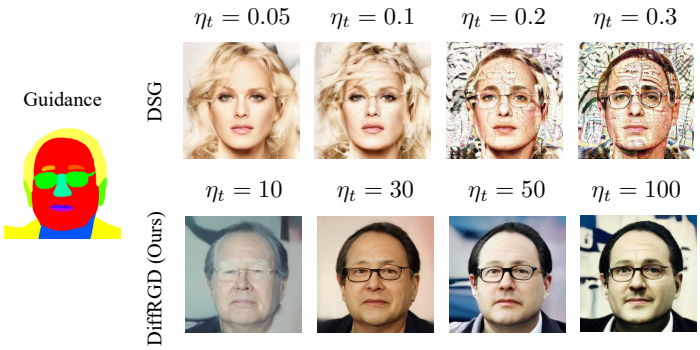

Fig. 7: Comparison of different guidance strengths for segmentation map-guided conditional generation.

Table 6: Influence of guidance strength η_t on segmentation map conditional generation with DSG and DiffRGD.

	DSG				DiffRGD			
η_t	0.05	0.1	0.2	0.3	10	30	50	100
mIoU ↑	0.53	0.76	0.90	0.93	0.74	0.82	0.84	0.86
FID ↓	95.69	110.00	157.82	248.77	80.73	89.03	93.47	108.33

5 Conclusion

To address the sample quality degradation arising from the failure to respect the stepwise Gaussian latent structure of diffusion models, we propose a novel inference-time guidance method, *DiffRGD*. Our method constrains updates to a latent-distribution-induced spherical manifold and solves each step via Riemannian Gradient Descent, maintaining distributional fidelity while enabling effective conditioning. On image restoration and conditional generation tasks, DiffRGD matched or surpassed prior inference-time methods, improving task metrics while preserving sample quality. These results indicate that geometry-consistent guidance is a viable alternative to directly applying the gradient in the ambient space or in a restrictive proposed manifold. In sum, respecting diffusion's latent distribution yields a simple, plug-and-play guidance with strong practical payoff.

Acknowledgements

This research is supported by National Science and Technology Council, Taiwan (R.O.C), under the grant number of NSTC-113-2115-M-003-012-MY2, NSTC-113-2221-E-002-201, NSTC-113-2221-E-007-105-MY3, NSTC-114-2221-E-001-004, NSTC-114-2221-E-002-182-MY3, NSTC-114-2634-F-001-001-MBK, NSTC-114-2634-F-002-004, and Academia Sinica under the grant number of AS-IAIA-114-M10. We sincerely thank Kang-Yang Huang for the discussions and valuable feedback during the rebuttal stage. We also thank Yu-Hsuan Lin and Hsien-Yueh Huang for their continuous encouragement throughout this project.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2008) 2, 4, 5
2. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, R., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: International Conference on Learning Representations (ICLR) (2024) 2, 3
3. Chan, S.H., Wang, X., Elgendy, O.A.: Plug-and-play admm for image restoration: Fixed-point convergence and applications. IEEE Transactions on Computational Imaging (2016) 4
4. Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion posterior sampling for general noisy inverse problems. In: International Conference on Learning Representations (ICLR) (2023) 1, 2, 3, 6, 9, 10, 11, 12
5. Chung, H., Sim, B., Ryu, D., Ye, J.C.: Improving diffusion models for inverse problems using manifold constraints. In: Advances in Neural Information Processing Systems (NeurIPS) (2022) 2
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 11
7. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 13
8. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2021) 1, 3, 9
9. Efron, B.: Tweedie's formula and selection bias. Journal of the American Statistical Association (2011) 4
10. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: International Conference on Machine Learning (ICML) (2024) 3
11. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Computers & mathematics with applications (1976) 4
12. He, Y., Murata, N., Lai, C.H., Takida, Y., Uesaka, T., Kim, D., Liao, W.H., Mitsufuji, Y., Kolter, J.Z., Salakhutdinov, R., et al.: Manifold preserving guided diffusion. In: International Conference on Learning Representations (ICLR) (2024) 1, 2, 4, 6, 9, 10, 11
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems (NIPS) (2017) 11
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 1, 3, 4
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS Workshop (NeurIPSW) on Deep Generative Models and Downstream Applications (2021) 1, 3
16. Jain, A.K.: Fundamentals of digital image processing. Englewood Cliffs, NJ: Prentice Hall (1989) 11
17. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (ICLR) (2018) 14

18. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: Advances in Neural Information Processing Systems (NeurIPS) (2022) 3
19. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 9
20. Kawar, B., Elad, M., Ermon, S., Song, J.: Denoising diffusion restoration models. Advances in Neural Information Processing Systems (NeurIPS) (2022) 2, 9
21. Kim, M., Levy, A., Wetzstein, G.: Dual ascent diffusion for inverse problems. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026) 9
22. Labs, B.F.: Flux. https://github.com/black-forest-labs/flux (2024), accessed: 2024-11-14 3
23. Labs, B.F.: Official weights of FLUX.1 dev. https://huggingface.co/black-forest-labs/FLUX.1-dev (2024), accessed: 2024-11-14 3
24. Lai, C.H., Song, Y., Kim, D., Mitsufuji, Y., Ermon, S.: The principles of diffusion models. arXiv preprint arXiv:2510.21890 (2025) 1
25. Liao, J.W., Wang, W., Wang, T.S., Peng, L.X., Weng, J.H., Chou, C.F., Chen, J.C.: DiffQRCoder: Diffusion-based aesthetic QR code generation with scanning robustness guided iterative refinement. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2025) 3
26. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2
27. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (ICLR) (2022) 13
28. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: International Conference on Machine Learning (ICML) (2022) 3
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems (NIPS) (2019) 9
30. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., Wolf, T.: Diffusers: State-of-the-art diffusion models (2022), https://github.com/huggingface/diffusers 2, 9
31. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 3, 12
32. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2023) 2
33. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning (ICML) (2015) 1, 3

34. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (ICLR) (2021) 2, 3, 4
35. Song, J., Vahdat, A., Mardani, M., Kautz, J.: Pseudoinverse-guided diffusion models for inverse problems. In: International Conference on Learning Representations (ICLR) (2023) 2
36. Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.Y., Kautz, J., Chen, Y., Vahdat, A.: Loss-guided diffusion models for plug-and-play controllable generation. In: International Conference on Machine Learning (ICML) (2023) 2, 3
37. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 4
38. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (ICLR) (2020) 1, 3, 4
39. Tibshirani, R.J., Efron, B.: An introduction to the bootstrap. Monographs on statistics and applied probability (1993) 11, 14
40. Tikhonov, A.N.: Solutions of ill posed problems. SIAM Review (1977) 9
41. Tsai, Y.Y., Chen, F.C., Chen, A.Y., Yang, J., Su, C.C., Sun, M., Kuo, C.H.: Gda: Generalized diffusion for robust test-time adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3
42. Voynov, A., Aberman, K., Cohen-Or, D.: Sketch-guided text-to-image diffusion models. In: ACM SIGGRAPH Conference proceedings (2023) 3
43. Wang, X., Chan, S.H.: Parameter-free plug-and-play admm for image restoration. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2017) 4
44. Wang, Y., Yu, J., Zhang, J.: Zero-shot image restoration using denoising diffusion null-space model. In: International Conference on Learning Representations (ICLR) (2023) 2, 9
45. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing (2004) 11
46. Wu, G., Liu, X., Jia, J., Cui, X., Zhai, G.: Text2qr: Harmonizing aesthetic customization and scanning robustness for text-guided qr code generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3
47. Xiang, X., Liu, D., Yang, X., Zhu, Y., Shen, X., Allebach, J.P.: Adversarial open domain adaptation for sketch-to-photo synthesis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2022) 13
48. Xu, X., Chi, Y.: Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction. In: Advances in Neural Information Processing Systems (NeurIPS) (2024) 9
49. Yang, L., Ding, S., Cai, Y., Yu, J., Wang, J., Shi, Y.: Guidance with spherical gaussian constraint for conditional diffusion. In: International Conference on Machine Learning (ICML) (2024) 1, 2, 4, 9, 10, 11, 12, 13
50. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 13
51. Yu, J., Wang, Y., Zhao, C., Ghanem, B., Zhang, J.: FreeDoM: Training-free energy-guided conditional diffusion model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) 2, 3, 6, 9, 13

52. Zhang, B., Chu, W., Berner, J., Meng, C., Anandkumar, A., Song, Y.: Improving diffusion inverse problem solving with decoupled noise annealing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025) 9
53. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) 1, 3
54. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 11
55. Zhang, Y., Liu, Z., Li, Z., Li, Z., Clark, J.J., Si, X.: Decoupling training-free guided diffusion by admm. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025) 2, 4, 9, 10, 11, 13
56. Zhu, Y., Zhang, K., Liang, J., Cao, J., Wen, B., Timofte, R., 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 Workshop (CVPRW) (2023) 2, 9