CORRECTING DIFFUSION HALLUCINATIONS VIA UNCERTAINTY-GUIDED TRAJECTORY STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have emerged as state-of-the-art image generative models, yet they still suffer from *hallucinations*, i.e., unrealistic artifacts or implausible structures absent from the training data distribution. Prior work attributes this issue to *mode interpolation*, where the model generates samples that lie between nearby data modes but fall outside the actual data support. Motivated by this perspective, we propose a lightweight image editing method for correcting hallucinations in diffusion-generated images. The key idea is to steer the sampling trajectory along directions where the diffusion model's posterior mean estimator exhibits the highest uncertainty, thereby pushing the generation process away from unsupported, uncertain regions and toward nearby realistic modes, while largely preserving the original semantic content. Notably, our method requires neither backpropagation nor external supervision. Experiments across multiple datasets show that it consistently reduces hallucinations while remaining faithful to the original image, achieving a more favorable validity–fidelity trade-off than existing techniques.

1 INTRODUCTION

Diffusion models have emerged as a powerful class of deep generative models, with successful applications across images (Dhariwal & Nichol, 2021; Yu et al., 2026), audio (Jiang et al., 2026), video (Xi et al., 2026), and text (Xiong et al., 2026). By learning to reverse a gradual noising process, diffusion models enable sampling from a simple prior distribution and progressively transform noise into data samples that follow the target data distribution (Ho et al., 2020). In image synthesis, diffusion models are particularly appealing due to their high sample fidelity (Chen et al., 2026b), strong controllability under guidance Starodubcev et al. (2026), rich semantic alignment with text prompts (Lee et al., 2026), and flexible trade-off between generation quality and computational cost (Cao et al., 2025).

Despite these advantages, diffusion models remain susceptible to hallucinations (Liu et al., 2026; Lu et al., 2024; Betti et al., 2026), namely the generation of unrealistic artifacts or implausible structures that are absent from the training data distribution (e.g., extra fingers, distorted limbs, or malformed objects). Such hallucinations undermine the reliability of diffusion-based image generation, particularly in downstream applications that require structural correctness or semantic faithfulness, such as medical image synthesis (Tian et al., 2025; Kim et al., 2024).

Prior work has explored various strategies to mitigate hallucinations by introducing external supervision or additional optimization procedures. For example, incorporating explicit structural constraints (e.g., human pose or depth maps) to correct malformed regions (Lu et al., 2024; Weng et al., 2024; Narasimhaswamy et al., 2024), or utilizing extra reference images (Samuel et al., 2024; Zhang et al., 2025; Zhu et al., 2024). However, these approaches typically rely on auxiliary knowledge sources, incur substantial computational overhead, or require additional training stages. As a result, they are not easily applicable as lightweight post hoc corrections to existing, frozen diffusion models. This motivates the following research question:

Can we correct hallucinations using only a frozen diffusion model itself, while introducing only minimal changes to the original semantic content of the generated image?

Under the premise that diffusion models have already internalized the knowledge required to generate realistic images, we focus on intervening in the sampling process to correct hallucinated outputs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

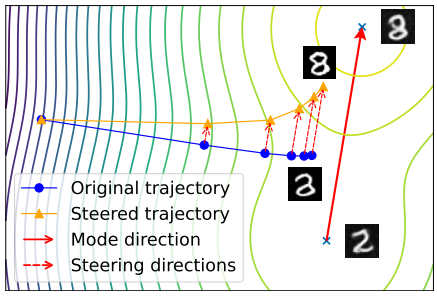


Figure 1: An illustrative example of steering a sampling trajectory out of a mode-interpolated region to correct hallucination. The hallucinated digit lies in the interpolated region between modes "2" and "8". By steering the trajectory along the direction connecting these modes, the edited image moves closer to mode "8", thereby alleviating the hallucination.

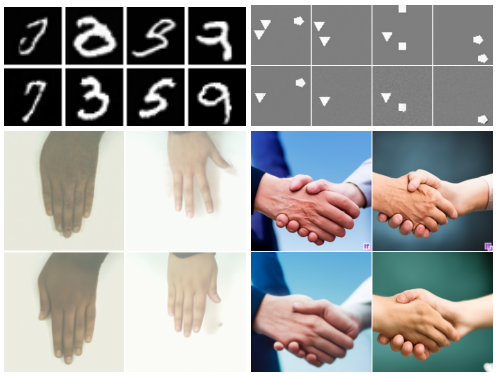


Figure 2: Qualitative results of our UTS algorithm. The top row shows the original samples, and the bottom row shows the edited results. UTS corrects hallucinated artifacts (e.g., unrecognizable digit, repeated shapes, and malformed hand structures) while largely preserving the original semantic content.

Designing such inference-time interventions requires understanding how hallucinations arise during the sampling process. Prior work (Aithal et al., 2024) attributes hallucinations to *mode interpolation*: when approximating the true data distribution, diffusion models may oversmooth regions between nearby high-density modes. As a result, samples can fall into these interpolated low-density regions that lie outside the support of the real distribution, leading to hallucinations. Notably, trajectories associated with hallucinated samples often exhibit elevated instability, as the denoising dynamics are influenced by competing attractions from neighboring modes. This instability provides a principled signal for detecting hallucinated images.

Motivated by this, we propose **Uncertainty-guided Trajectory Steering (UTS)**, an *inference-time* algorithm that corrects hallucinated artifacts by explicitly modifying the sampling trajectory of a frozen diffusion model. Given a hallucinated image, we first approximate its denoising trajectory via DDIM inversion (Song et al., 2021), thereby recovering the intermediate latent states that produced the image. We then inject carefully designed steering vectors at selected timesteps to nudge these intermediate states toward nearby high-density modes responsible for mode interpolation. The steering directions are constructed to approximate the directions connecting such nearby modes, which are hypothesized to give rise to hallucinations. By doing so, the modified trajectory is encouraged to converge to a realistic mode with similar semantic content but reduced hallucination.

Fig. 1 illustrates the core idea of our method using a two-mode Gaussian mixture. The original sampling trajectory (shown with circular markers) terminates in an interpolated low-density region between the two modes, resulting in a hallucinated sample. By applying steering vectors aligned with the direction connecting the two modes, the modified trajectory (shown with triangle markers) is guided away from the interpolated region and converges toward a nearby high-density mode, yielding a more realistic outcome. We further provide theoretical analysis to justify the proposed design: under a simple Gaussian mixture setting, we show that our method can recover the direction connecting nearby modes, and that each steering step can be interpreted as a form of implicit classifier guidance (Dhariwal & Nichol, 2021), where nearby data modes play the role of latent classes, and steering along the mode direction biases the sampling trajectory toward one such mode, analogous to increasing the likelihood of a target class.

We validate our approach across a diverse set of datasets, including MNIST digits, synthetic shape datasets with predefined structural rules, real-world hand images, and high-resolution Text-to-Image (T2I) diffusion models. Fig. 2 presents qualitative results across datasets. The top row shows the original images, and the bottom row shows the edited outputs. Our method corrects hallucinated artifacts while largely preserving the original semantic content.

To systematically characterize hallucination correction, we evaluate each method along two complementary dimensions: *validity* and *fidelity*, which respectively capture the extent to which hallucinated artifacts are removed and original semantic content is preserved. Since neither concept is directly mea-

surable, we employ automatically computed proxy metrics for quantitative evaluation. We compare the trade-off hypervolume (Chen et al., 2026a; Zitzler & Thiele, 1999) of our method against existing baselines. Across most settings, our approach consistently achieves a more favorable trade-off and better validity, substantially reducing hallucinations while minimally altering the original content. More related work is discussed in Appendix E. Our main contributions are as follows:

- We analyze causes of hallucinations in diffusion-based image synthesis through the lens of mode interpolation, and propose *Uncertainty-guided Trajectory Steering* (UTS), an inference-time algorithm that removes hallucinated artifacts with minimal modification to the original content.
- Under a Gaussian mixture setting, we provide theoretical justification showing that UTS can identify directions associated with hallucination-inducing modes, and that each steering step admits an interpretation as implicit classifier guidance.
- We evaluate UTS on a diverse range of hallucinated images, from simple geometric shapes to text-aligned images, and show that it consistently achieves a more favorable validity–fidelity trade-off than existing baselines.

2 Preliminaries: diffusion model

We first review Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020). Given a sample drawn from the real distribution, $\boldsymbol{x}_0 \sim q(\boldsymbol{x})$, DDPM defines a *forward process* that gradually corrupts the data by adding Gaussian noise over a sequence of timesteps. By applying the reparameterization trick to the Markov transition process, we can sample noisy observations at timestep t by:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_t, \tag{1}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\bar{\alpha}_t$ are predefined schedule coefficients. The generative procedure learns a *reverse process* that iteratively denoises a sample initialized from pure Gaussian noise $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. At each timestep, the model approximates the reverse conditional distribution $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, ultimately producing samples that follow $q(\boldsymbol{x})$. A denoising diffusion model $\boldsymbol{\epsilon}_\theta$ is trained to predict the injected noise $\boldsymbol{\epsilon}_t$ from the noisy input $\boldsymbol{x}_t$ and timestep t, by minimizing the following objective:

$$\mathcal{L} = \mathbb{E}_{t\sim[1,T],\boldsymbol{x}_0,\boldsymbol{\epsilon}_t}\Big[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\|^2\Big],$$

where $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ denotes the model's noise prediction.

Sampling proceeds by iteratively applying the learned reverse diffusion dynamics. While DDPM sampling follows a stochastic trajectory, Denoising Diffusion Implicit Models (DDIM) (Song et al., 2021) define a deterministic sampling procedure that preserves the same marginal distributions as DDPM. Given a trained DDPM denoiser $\boldsymbol{\epsilon}_\theta$, the DDIM sampling update is

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t). \tag{2}$$

Compared to DDPM, DDIM can generate high-quality samples using fewer sampling steps. Assuming $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \approx \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1}, t)$, DDIM admits an approximate inversion procedure that maps a given image back to a noisy latent:

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

The resulting noisy latent approximately reconstructs the original image under DDIM sampling and provides a latent space for controlled edits (Huang et al., 2025). Further background details are provided in Appendix A.

3 Observations and Design Motivation

Our goal is to correct hallucinations in diffusion-generated images while preserving their original semantic content. Given a **frozen** diffusion model $\boldsymbol{\epsilon}_\theta$ and a hallucinated image $\boldsymbol{x}$ sampled from it, we seek to identify steering vectors that guide the sampling process toward a realistic data mode, yielding an edited image $\tilde{\boldsymbol{x}}$ that removes hallucinated artifacts with minimal deviation from the original image,



(a) $\hat{\boldsymbol{x}}_0^{(t)}$ computed from ground-truth DDPM trajectory

(b) $\hat{\boldsymbol{x}}_0^{(t)}$ computed from independently sampled $\boldsymbol{x}_t$

(c) Principal components of $\hat{\boldsymbol{x}}_0^{(t)}$ computed from $\boldsymbol{x}_t$

Figure 3: Trajectories of PMP and principal components along the denoising process, with the upper-left panel corresponding to $t = 16$, and the sequence progressing to the bottom-right panel at $t = 1$. (a) Visualizes $\hat{\boldsymbol{x}}_0^{(t)}$ computed from the ground-truth DDPM trajectories that produce the hallucinated image. (b) Visualizes $\hat{\boldsymbol{x}}_0^{(t)}$ computed from independently sampled $\boldsymbol{x}_t$ conditioned on the same final image. (c) Visualizes the first principal components computed from 20 independently sampled $\boldsymbol{x}_t$ at each timestep. Comparing (a) and (b), we observe substantial variability at the same timestep (e.g., $t = 13$), which motivates the use of timestep-conditioned uncertainty as a signal for identifying directions associated with hallucinations. Panel (c) further shows that uncertainty in the PMP at intermediate timesteps is directly related with hallucination artifacts.

particularly in background regions and overall semantic structure. This section presents the key insights and technical components that form the foundation of our approach.

Hallucinations via Mode Interpolation. We begin with preliminary analyses to illustrate how hallucinations emerge during diffusion sampling and to motivate our method design. Specifically, we examine the behavior of predicted clean images along the denoising trajectory for hallucinated samples. Given a trained denoiser $\boldsymbol{\epsilon}_\theta$ and an intermediate latent $\boldsymbol{x}_t$, we can estimate the posterior mean $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$ of the clean data conditioned on $\boldsymbol{x}_t$ via

$$\hat{\boldsymbol{x}}_0^{(t)} \stackrel{\text{def}}{=} f(\boldsymbol{x}_t) = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{3}$$

This posterior mean predictor (PMP) provides a deterministic estimate of the underlying clean image corresponding to the noisy latent $\boldsymbol{x}_t$, and has been shown to encode rich semantic and structural information about the data (Chen et al., 2024; Manor & Michaeli, 2023; Aithal et al., 2024). Given a hallucinated image $\boldsymbol{x}_0$, we collect the corresponding ground-truth DDPM sampling trajectory $\{\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, \ldots, \boldsymbol{x}_1\}$ that produces this output. At each timestep t, we compute the predicted clean image $\hat{\boldsymbol{x}}_0^{(t)}$ using the PMP. Fig. 3a visualizes these predicted $\hat{\boldsymbol{x}}_0^{(t)}$ across timesteps, with the upper-left panel corresponding to $t = 16$, and the sequence progressing to the bottom-right panel at $t = 1$.

Notably, the predicted clean images at different timesteps may exhibit distinct modes, suggesting that the sampling trajectory interpolates between nearby data modes before converging to a final hallucinated image. For example, in Fig. 3a, the predictions at $t = 13$ and $t = 12$ transition between two plausible hand configurations with different orientations. This phenomenon aligns with the mode interpolation explanation of hallucinations.

Timestep-Conditioned Uncertainty. Aithal et al. (2024) leverage temporal instability of the PMP trajectory as a signal for hallucination detection. However, temporal variance observed along the denoising trajectory conflates *two fundamentally different sources of uncertainty*. The first arises from mode interpolation, reflecting ambiguity among competing nearby modes and constituting the primary cause of hallucinations. The second is intrinsic to denoising process itself: early timesteps correspond to heavily corrupted inputs and thus naturally exhibit higher variance, which gradually diminishes as denoising progresses and image structure becomes increasingly well defined. As a result, temporal variance aggregates both effects, making it difficult to disentangle instability induced by mode interpolation from that due to the inherent progression of denoising. This motivates the need for a more localized measure that isolates uncertainty specifically attributable to mode interpolation.

To address this, we propose to characterize uncertainty *conditioned on the timestep*. For a fixed generated image $\boldsymbol{x}_0$ and a given timestep t, we resample multiple latent variables $\boldsymbol{x}_t$ according to Eq. (1), each of which corresponds to a valid DDPM sampling trajectory that starts from a different noise realization but converges to the same final image. Such alternative, or *counterfactual*, trajectories have been explored in prior work as a means of extracting semantic information for training-free image editing (Wu & De la Torre, 2023; Huberman-Spiegelglas et al., 2024). By jointly considering these counterfactual trajectories, we can examine the variability of the predicted clean image $\hat{\boldsymbol{x}}_0$ at a fixed timestep, thereby isolating uncertainty conditioned on t. Under the mode interpolation hypothesis, this timestep-conditioned uncertainty is expected to concentrate along directions connecting nearby data modes, reflecting the competing influence of multiple plausible modes during sampling. Fig. 3b visualizes resampled latents at different timesteps. Compared with Fig. 3a, we observe variability across counterfactual trajectories at the same timestep, e.g., at $t = 13$. This observation motivates our use of timestep-conditioned uncertainty as a signal for identifying and correcting hallucinated structures.

Identifying Mode Directions. Building on the above observations, we next seek to identify the *mode direction*, i.e., the dominant direction connecting nearby data modes whose interpolation gives rise to hallucinations. Our goal is to steer the sampling trajectory along this direction so that it converges toward a single plausible mode, thereby escaping the interpolated region and producing an edited image without hallucinated artifacts.

Concretely, for a fixed image $\boldsymbol{x}_0$ and timestep t, we sample multiple latent variables $\boldsymbol{x}_t$ according to Eq. (1). For each sampled $\boldsymbol{x}_t$, we compute corresponding predicted clean images via PMP, yielding a collection of $\hat{\boldsymbol{x}}_0^{(t)}$. We then perform principal component analysis (PCA) on these predictions, and take the leading principal component as an estimate of the mode direction at timestep t. Fig. 3c visualizes the resulting principal components across timesteps, where we sample 20 latents per timestep. For visualization purposes, each image is independently normalized using min–max scaling.

Two key observations emerge. First, the uncertainty captured by the PMP exhibits a coarse-to-fine progression along the denoising trajectory: principal components at early timesteps reflect global and semantically ambiguous variations, whereas those at later timesteps become increasingly localized, consistent with prior observations of diffusion dynamics (Ho et al., 2020; Aithal et al., 2024). Second, variance in the PMP at intermediate timesteps highlights uncertainty that is directly associated with hallucinations. For example, in Fig. 3c, the white and black regions in red boxes correspond to the model's uncertainty about whether to include the little finger (ultimately missing in the final hallucinated output), indicating competition among different hand configurations. Together, these findings support the use of PMP-based principal directions as effective signals for identifying and removing hallucinations. To confirm that the first principal component provides the most meaningful and reliable steering direction, we visualize higher-order principal components in Appendix D.6; the results show that these components capture less reliable structure for hallucination correction than the leading component.

Theoretical Analysis. To justify why the leading principal component captures the mode direction, we provide a theoretical analysis in a stylized setting. Consider a data distribution given by a mixture of two Gaussians,

$$\boldsymbol{x}_0 \sim \pi\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{I}) + (1-\pi)\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{I}),$$

where $\pi \in (0, 1)$, and noisy observations $\boldsymbol{x}_t$ are generated according to Eq. (1). Given a realization of $\boldsymbol{x}_t$, we compute the predicted clean image $\hat{\boldsymbol{x}}_0^{(t)}$ using the PMP. Assume the trained $\boldsymbol{\epsilon}_\theta$ matches the ground-truth distribution. The following theorem shows that the principal direction of the covariance of $\hat{\boldsymbol{x}}_0^{(t)}$ aligns with the direction connecting the two modes, defined as $\boldsymbol{\delta} \coloneqq \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$.

Theorem 3.1 (Identifying the mode direction)**.** *The leading eigenvector of* $\mathrm{Cov}(\hat{\boldsymbol{x}}_0^{(t)})$ *is parallel to* $\boldsymbol{\delta}$.

The proof of Theorem 3.1 is deferred to Appendix B. Although the assumed data distribution is intentionally simple, it can be viewed as a local approximation of more complex scenarios with multiple competing modes (beyond the two-mode case). Aithal et al. (2024) observe that interpolation occurs only between *nearest-neighbor modes*. This motivates the following reasoning: as the denoising process proceeds, the sample is progressively attracted toward a single mode and the primary source of hallucination is mode interpolation between the attracted mode and its nearest neighbor. The principal components increasingly reflect the dominant axis between the two locally competing modes, which we will show empirically sufficient to suppress hallucination.

4 PROPOSED ALGORITHM

Building on the observations and theoretical insights from Section 3, we now introduce our inference-time algorithm for correcting hallucinations in diffusion-generated images.

Steering Latents. Having identified the mode direction $\boldsymbol{\delta}_t$ at timestep t, we next seek to steer the intermediate latent state $\boldsymbol{x}_t$ so that the corresponding predicted clean image $\hat{\boldsymbol{x}}_0^{(t)}$ moves along this direction. Since the predicted clean image can be obtained by PMP, $\hat{\boldsymbol{x}}_0^{(t)} = f(\boldsymbol{x}_t)$ (Eq. (3)), we aim to find a steering vector $\Delta\boldsymbol{x}$ such that perturbing $\boldsymbol{x}_t$ induces a controlled displacement of $\hat{\boldsymbol{x}}_0^{(t)}$ along $\boldsymbol{\delta}_t$, i.e.,

$$f(\boldsymbol{x}_t + \Delta\boldsymbol{x}) \approx f(\boldsymbol{x}_t) + \lambda\boldsymbol{\delta}_t,$$

where λ controls the steering strength.

Algorithm 1 Uncertainty-guided Trajectory Steering (UTS)

Input: Pre-trained diffusion model $\boldsymbol{\epsilon}_\theta$, hallucinated image $\boldsymbol{x}_0$ sampled from $\boldsymbol{\epsilon}_\theta$, starting timestep $\bar{T}$, ending timestep $\bar{t}$.

```
Reversing x_0 to x_T̄ by DDIM Inversion
for t = T̄ to 0 do
    if t > t̄ then
        Compute x̂_0^(t) from x_t by Eq. (3)
        Sample a batch of x̃_t = √ᾱ_t x̂_0^(t) + √(1−ᾱ_t) ε, ε ~ N(0, I).
        Compute x̂_0^(t)~ for each x̃_t by Eq. (3).
        Compute the leading principal component δ_t for the x̂_0^(t)~ batch.
        γ ← sign(r^⊤ ∇_{x_t} log q(x_t))
        x̃_t ← x_t + γλ√ᾱ_{t−1} δ_t
    end if
    DDIM Inference step by Eq. (2) to get x_{t−1}.
end for
Output: Edited image x̃_0
```

To obtain a tractable expression for steering vector $\Delta\boldsymbol{x}$, we make a mild assumption that the denoiser output is locally stable with respect to small perturbations of the input, i.e., $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \approx \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t + \Delta\boldsymbol{x}, t)$. This approximation is closely related to the assumption underlying DDIM inversion, as discussed in Section 2. We can then derive an approximate closed-form solution for the steering vector using (3):

$$\Delta\boldsymbol{x} \approx \lambda\sqrt{\bar{\alpha}_t}\boldsymbol{\delta}_t.$$

Theoretical Interpretation via Classifier Guidance. We now provide an interpretation of our per-step steering rule through the lens of *classifier guidance* (Dhariwal & Nichol, 2021). In standard classifier guidance, the sampling trajectory of a diffusion model is biased toward a desired class by incorporating label information y through an auxiliary classifier. Specifically, given a classifier that estimates the conditional distribution $p(y \mid \boldsymbol{x}_t)$, the denoising prediction is modified by adding a gradient term that steers the sample toward regions of higher class probability:

$$\bar{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + w\sqrt{1-\bar{\alpha}_t}\,\nabla_{\boldsymbol{x}_t}\log p(y \mid \boldsymbol{x}_t), \tag{4}$$

where w controls the strength of the guidance.

In our setting, each data mode can be interpreted as a latent class, and hallucinations arise when the sampling trajectory lies in regions where multiple such latent classes compete. The direction connecting nearby modes therefore plays a role analogous to the gradient of a latent class likelihood. Our steering rule, which biases the trajectory along this mode direction, can thus be viewed as a form of *implicit classifier guidance*. Under the same Gaussian mixture setting as in Theorem 3.1, we can formalize this connection through the following theorem.

Theorem 4.1 (Equivalence with classifier guidance)**.** *Steering $\boldsymbol{x}_t$ with vector $\Delta\boldsymbol{x} \approx \lambda\sqrt{\bar{\alpha}_t}\boldsymbol{\delta}_t$ is equivalent to applying classifier guidance in Eq.* (4)*, with guidance strength*

$$w = \pm\frac{\lambda\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t(1-\bar{\alpha}_t)(1-\bar{\alpha}_{t-1})} - \sqrt{\bar{\alpha}_{t-1}}(1-\bar{\alpha}_t)(1-\mathcal{P}(\boldsymbol{x}_t, k))},$$

where scalar $\mathcal{P}(\boldsymbol{x}_t, k) \coloneqq 1 - p(k \mid \boldsymbol{x}_t)$, $k \in \{0, 1\}$ denotes the mode index of the Gaussian mixture that generates $\boldsymbol{x}_t$, where the sign determines which latent class the guidance steers toward.

The proof of Theorem 4.1 is provided in Appendix C. This result shows that our method can be interpreted as implicitly performing classifier guidance toward a nearby mode, which acts as a latent class responsible for generating the sample. Notably, this guidance emerges without training an explicit classifier, relying instead on the uncertainty structure encoded in the posterior mean predictor.

Table 1: Average fidelity (LPIPS, lower is better), validity (dataset-specific score, higher is better), each averaged over every hyperparameter setting of that method, and validity–fidelity trade-off hypervolume (HV, higher is better). Bold marks the best validity and HV per dataset. Our UTS method achieves the best trade-off on 3 of 4 datasets and best validity on all datasets.

Method	MNIST			Simple Shapes			Hands			Shaking Hands		
	Fid.↓	Val.↑	HV↑	Fid.↓	Val.↑	HV↑	Fid.↓	Val.↑	HV↑	Fid.↓	Val.↑	HV↑
DI	0.014	4.535	0.5193	0.135	0.001	0.0005	0.120	0.072	0.0163	0.013	0.215	0.0589
RTS	0.106	5.074	0.6607	0.283	0.173	0.1280	0.119	0.071	0.0167	0.016	0.198	0.0598
CAS	0.079	5.287	0.8125	0.318	0.067	0.0857	0.154	0.071	0.0131	0.028	0.207	0.0556
DG	0.161	4.617	0.6726	0.493	0.578	**0.1686**	0.160	0.009	0.0067	0.316	0.083	0.0079
UTS (Ours)	0.168	**7.181**	**0.8565**	0.546	**0.944**	0.1664	0.154	**0.184**	**0.0332**	0.128	**0.231**	**0.0687**

Uncertainty-guided Trajectory Steering. We now describe the complete algorithm for uncertainty-guided trajectory steering (Algorithm 1). Given a hallucinated image $\boldsymbol{x}_0$ generated by a frozen diffusion model $\boldsymbol{\epsilon}_\theta$, a target inversion timestep $\bar{T}$, and an early-stopping timestep $\bar{t}$, we first apply DDIM inversion to map $\boldsymbol{x}_0$ to a noisy latent $\boldsymbol{x}_{\bar{T}}$. Starting from $\boldsymbol{x}_{\bar{T}}$, we iteratively perform steering followed by DDIM sampling in a step-wise manner. At each timestep t, we dynamically estimate uncertainty and apply steering. Specifically, we first compute the predicted clean image $\hat{\boldsymbol{x}}_0^{(t)}$ using the PMP (Line 4). Conditioned on $\hat{\boldsymbol{x}}_0^{(t)}$, we then resample multiple latent states $\widetilde{\boldsymbol{x}_t}$ by replacing $\boldsymbol{x}_0$ with $\hat{\boldsymbol{x}}_0^{(t)}$ in Eq. (1) (Line 5), and compute the corresponding PMP predictions $\widetilde{\hat{\boldsymbol{x}}_0^{(t)}}$ for each $\widetilde{\boldsymbol{x}_t}$ (Line 6). Using these predictions, we perform principal component analysis to obtain the leading principal direction $\boldsymbol{\delta}_t$, which serves as an estimate of the mode direction at timestep t (Line 7).

To determine the orientation of the steering, we evaluate the alignment between the mode direction $\boldsymbol{\delta}_t$ and the score function $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t)$. The score function is approximated using the learned denoiser $\boldsymbol{\epsilon}_\theta$, leveraging the connection between DDPMs and score-based models (see Appendix A). We select the steering sign as $\gamma = \operatorname{sign}\Big(\boldsymbol{\delta}_t{}^\top \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t)\Big)$, ensuring that the steering is applied in the direction of increasing likelihood along the mode direction. The latent state is then updated by

$$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \gamma \lambda \sqrt{\bar{\alpha}_t}\, \boldsymbol{\delta}_t,$$

where λ controls the steering magnitude. Because hallucinations are associated with low-likelihood, interpolated regions, this update biases the latent toward higher-density regions of the data distribution, thereby steering it away from hallucinated artifacts. After applying the steering update, we proceed with a standard DDIM sampling step to advance the latent state to the next timestep (Line 11).

Importantly, our framework allows steering to be applied only within a selected window of timesteps. By adjusting the starting timestep $\bar{T}$ and the stopping timestep $\bar{t}$, we can restrict intervention to stages of the sampling process that are most critical to the formation of hallucinations, while leaving the remaining steps unchanged to better preserve the original content. Moreover, the proposed algorithm can be applied with an arbitrary number of DDIM sampling steps.

5 Experiments

Datasets. We evaluate our method across datasets of increasing complexity. For unconditional image generation, we train DDPM models on three datasets and analyze the resulting synthetic samples. We further assess our method on images generated by a pretrained text-to-image diffusion model. Hallucinated samples are identified automatically for every dataset, as state below. All editing experiments are conducted on RTX A5000 GPUs. Additional details are in Appendix D.

(i) MNIST (LeCun et al., 2010). We identify hallucinated samples as digits with the lowest Inception Score (IS) (Salimans et al., 2016) produced by a pretrained classifier with 99% accuracy, using generative quality as a proxy for structural plausibility. We select the 10,000 images with the lowest IS among 100,000 synthetic samples as our evaluation set. We define *validity* as IS after editing, and measure *fidelity* using LPIPS (Zhang et al., 2018), which quantifies perceptual similarity between the original and edited images.

(ii) Simple Shapes (Aithal et al., 2024). Following Aithal et al. (2024), we generate synthetic training images containing geometric shapes (triangle, square, pentagon), with the constraint that each image



(a) Simple Shapes (b) Hands (c) Shaking Hands

Figure 4: Qualitative results on the Simple Shapes, Hands, and Shaking Hands datasets. From top to bottom, rows correspond to DI, UTS, CAS, and DG. Our UTS algorithm corrects hallucinated artifacts and malformed hand structures without any external supervision.

contains at most one instance of each shape. Hallucinations in synthetic images are detected as repeated shapes or without any shape using OpenCV. We sample 100,000 synthetic images, of which 14,094 are identified as hallucinated. An edited image is considered *valid* if shape exists and no shape is repeated, and satisfies *fidelity* if the original shape types are preserved.

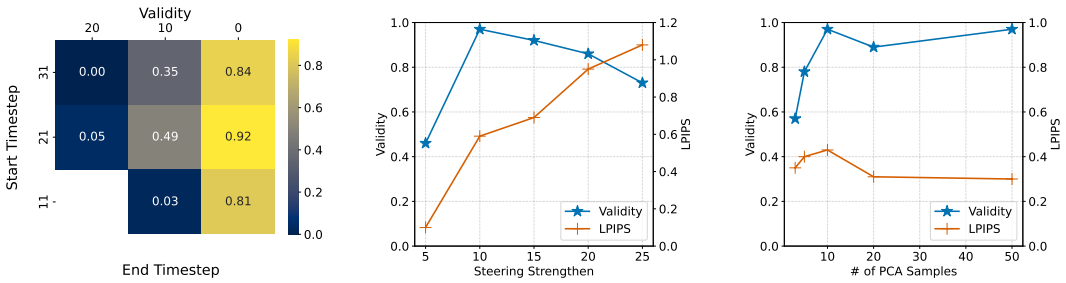
(iii) Hands (Afifi, 2019). The training dataset consists of photographs of human hands against a uniform white background. We sample 10,000 synthetic images and use MediaPipe Hands (Zhang et al., 2020) to automatically detect 2,000 hallucinated samples with the lowest hand-detection confidence score returned by the detector. *Validity* is measured by MediaPipe's detection confidence on the edited image, and *fidelity* is measured using LPIPS.

(iv) Shaking Hands. We use the prompt *"Shaking hands"* to generate 10,000 synthetic images on DeepFloyd-IF (DeepFloyd). Similar to Hands, we use MediaPipe Hands (Zhang et al., 2020) to select 1,000 hallucinated samples with the lowest confidence score. *Validity* is measured by MediaPipe's detection confidence, and *fidelity* is measured using LPIPS.

Our dataset selection follows a progressive difficulty: we begin with datasets where hallucinations can be automatically detected, and extend to more challenging real-world settings where hallucination is instead assessed via a trained plausibility model. Our evaluation scale is consistent with prior work in this area (Tian et al., 2025; Aithal et al., 2024). Moreover, our work is the first to correct hallucinations in text-to-image diffusion models without external knowledge, representing a more challenging setting.

Baselines. We compare *UTS* against three baselines: **(i) DDIM Inversion (DI)** (Song et al., 2021). DI reverses an image to a noisy latent via DDIM inversion and reconstructs it using standard DDIM sampling. It can be viewed as our method without trajectory steering, isolating the effect of reconstruction. **(ii) Randomized Trajectory Steering (RTS).** RTS replaces the principal steering direction in UTS with a random vector of the same norm at each timestep, allowing us to evaluate the importance of principled mode directions versus arbitrary perturbations. **(iii) CAS** (Tian et al., 2025). CAS mitigates hallucinations by detecting high-risk sampling steps and modifying the update rule via robust optimization based on local curvature. Unlike UTS, it does not leverage PMP uncertainty. **(iv) Dynamic Guidance (DG)** (Triaridis et al., 2025a). DG uses guidance towards modes to mitigate hallucination. DG relies on external supervision to define and identify modes; we nonetheless include it as a baseline for comparison.

Results. For each algorithm, we iterate over a predefined grid of hyperparameters. When different algorithms share the same hyperparameter, we apply identical values to ensure a fair comparison. Additional experimental results are in Appendix D. Table 1 shows fidelity and validity averaged over every hyperparameter setting per method (see Table 2 for the grids). We additionally report the Hypervolume (HV) indicator (Chen et al., 2026a; Zitzler & Thiele, 1999) to summarize the validity-fidelity trade-off in a single scalar. HV measures the area of the validity-fidelity space dominated by each method's Pareto front relative to a shared reference point; higher is better. UTS consistently achieves the best validity, and achieves a more favorable validity–fidelity trade-off than



(a) Steering window (b) Steering strengthen (c) Number of PCA samples

Figure 5: Ablation studies on Simple Shapes showing the effect of (a) steering timestep window, (b) steering strength, and (c) number of PCA samples on validity and fidelity.

baselines on most datasets. While UTS does not correct all hallucinated images, validity should be interpreted in light of the inherent difficulty of the task: correcting structural hallucinations **without any external knowledge**. To our knowledge, this is the first work to evaluate hallucination correction on text-to-image diffusion models in a fully unsupervised setting, and UTS achieves competitive performance compared with baselines. Fig. 4 presents qualitative results on three datasets. UTS effectively corrects hallucinated artifacts such as redundant shapes or malformed fingers while largely preserving the original semantic content, without requiring any external supervision.

Appendix D.5 reports the inference time of UTS and baselines. Results show that UTS achieves inference time comparable to CAS across all datasets. Appendix D.7 investigates the validity-diversity interaction, showing validity gains do not sacrifice diversity. While our method is primarily designed for image editing, it can also be applied to hallucination prevention setting combined with hallucination detection methods (Aithal et al., 2024; Tian et al., 2025), where we sample new images from scratch and steer the trajectory on the fly. Appendix D.8 shows results applying UTS in the prevention setting with diverse sampler beyond DDIM.

Fig. 5 reports ablation studies on the Simple Shapes dataset, analyzing the effects of (a) the steering timestep window, (b) steering strength, and (c) the number of PCA samples. Fig. 5a shows that the choice of the steering window significantly affects performance. In particular, initiating steering at intermediate timesteps (between the 31st and 11th steps) yields the highest validity. As illustrated in Figure 3c, PMP uncertainty follows a coarse-to-fine progression over timesteps, with hallucination-related uncertainty concentrated in intermediate denoising stages. Restricting steering to this region is therefore most effective for hallucination correction. Moreover, we observe that single-step steering is less effective than multi-step steering, suggesting that hallucinations arise from accumulated effects across multiple timesteps rather than a single critical step. Fig. 5b demonstrates that increasing steering strength initially improves correction performance but eventually degrades it, while perceptual distance from the original image monotonically increases. This indicates the existence of an optimal steering magnitude that balances validity and fidelity. Fig. 5c shows that increasing the number of samples used for PCA estimation substantially improves validity initially and then saturates, with marginal gains in fidelity. More samples better capture the semantic uncertainty directions encoded in the PMP, enabling more precise hallucination correction while avoiding unnecessary modifications. However, sampling additional intermediate states increases computational cost. Empirically, steering from $\lceil 0.8T \rceil$ to $\lceil 0.15T \rceil$ with steering strength 10 and 10 PCA samples performs well across datasets.

6 Conclusions

Motivated by prior work attributing hallucinations to mode interpolation, we observe that uncertainty in the posterior pean predictor (PMP) provides informative signals about hallucinations. Building on this observation, we identify mode directions from PMP uncertainty and steer the sampling trajectories of hallucinated images toward nearby high-density modes, producing edited results with reduced hallucination. We provide theoretical justifications and empirical results of our algorithm. Our approach achieves a more favorable validity–fidelity trade-off across a wide range of datasets, from simple geometric shapes to complex text-aligned images.

AI USE STATEMENT

In this work, generative models are subject of our study. We used generative AI models to generate synthetic images to support data analysis, from simple geometric shapes to human hands. We have not used generative AI tools to help develop theoretical frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, or interpret results. Assisting with translation are not applicable to this work. Additionally, we used generative AI tools to draft and edit parts of the paper to improve readability, and to edit software code. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The proposed algorithm is fully specified in Section 4. Appendix D.1 details every dataset and models used; and Appendix D.3 lists the exact hyperparameter grid swept for each method and dataset. Code is provided as anonymized supplementary material and will be released publicly upon publication; MNIST, Hands, and DeepFloyd-IF are publicly available, and Simple Shapes is procedurally generated from the rules described in Appendix D.1.

REFERENCES

Mahmoud Afifi. 11k hands: gender recognition and biometric identification using a large dataset of hand images. *Multimedia Tools and Applications*, 2019. doi: 10.1007/s11042-019-7424-8. URL https://doi.org/10.1007/s11042-019-7424-8.

Sumukh K Aithal, Pratyush Maini, Zachary Lipton, and J Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation. *Advances in Neural Information Processing Systems*, 37:134614–134644, 2024.

Federico Betti, Lorenzo Baraldi, Lorenzo Baraldi, Rita Cucchiara, and Nicu Sebe. Hallucination early detection in diffusion models. *International Journal of Computer Vision*, 134(1):35, 2026.

Pu Cao, Feng Zhou, Qing Song, and Lu Yang. Controllable generation with text-to-image diffusion models: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Stanley Chan et al. Tutorial on diffusion models for imaging and vision. *Foundations and Trends® in Computer Graphics and Vision*, 16(4):322–471, 2024.

Dongliang Chen, Xinlin Zhuang, Junjie Xu, Luojian Xie, Zehui Wang, Jiaxi Zhuang, Haolin Yang, Liang Dou, Xiao He, Xingjiao Wu, et al. Apex: Learning adaptive priorities for multi-objective alignment in vision-language generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4922–4939, 2026a.

Hao Chen, Junyang Chen, Jinshan Pan, and Jiangxin Dong. Bridging fidelity-reality with controllable one-step diffusion for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30584–30594, 2026b.

Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspace in diffusion models for controllable image editing. *Advances in neural information processing systems*, 37:27340–27371, 2024.

DeepFloyd. GitHub - deep-floyd/IF — github.com. https://github.com/deep-floyd/IF. [Accessed 20-01-2026].

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Liangliang Cao, and Shifeng Chen. Diffusion model-based image editing: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12469–12478, 2024.

Yuxuan Jiang, Zehua Chen, Zeqian Ju, Yusheng Dai, Weibei Dou, and Jun Zhu. Controlaudio: Tackling text-guided, timing-indicated and intelligible audio generation via progressive diffusion modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1394–1413, 2026.

Seunghoi Kim, Chen Jin, Tom Diethe, Matteo Figini, Henry FJ Tregidgo, Asher Mullokandov, Philip Teare, and Daniel C Alexander. Tackling structural hallucination in image translation with local diffusion. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Seunghoi Kim, Henry FJ Tregidgo, Matteo Figini, Chen Jin, Sarang Joshi, and Daniel C Alexander. Tackling hallucination from conditional models for medical image reconstruction with dynamicdps. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 593–603. Springer, 2025.

Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Jaa-Yeon Lee, Byunghee Cha, Jeongsol Kim, and Jong Chul Ye. Aligning text to image in diffusion models is easier than you think. *Advances in Neural Information Processing Systems*, 38:157106–157136, 2026.

Xi Liu, Weiwei Sun, Zhou Ren, Chris Broaddus, Siyu Huang, and Laurent Guigues. Had: Hallucination-aware diffusion priors for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29781–29791, 2026.

Rui Lu, Runzhe Wang, Kaifeng Lyu, Xitai Jiang, Gao Huang, and Mengdi Wang. Towards understanding text hallucination of diffusion models via local generation bias. In *The Thirteenth International Conference on Learning Representations*, 2025.

Wenquan Lu, Yufei Xu, Jing Zhang, Chaoyue Wang, and Dacheng Tao. Handrefiner: Refining malformed hands in generated images by diffusion-based conditional inpainting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7085–7093, 2024.

Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. *arXiv preprint arXiv:2309.13598*, 2023.

Supreeth Narasimhaswamy, Uttaran Bhattacharya, Xiang Chen, Ishita Dasgupta, Saayan Mitra, and Minh Hoai. Handiffuser: Text-to-image generation with realistic hand appearances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2468–2479, 2024.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4695–4703, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Nikita Starodubcev, Daniil Pakhomov, Zongze Wu, Ilya Drobyshevskiy, Yuchen Liu, Zhonghao Wang, Yuqian Zhou, Zhe Lin, and Dmitry Baranchuk. Rethinking global text conditioning in diffusion transformers. *arXiv preprint arXiv:2602.09268*, 2026.

Yiqi Tian, Pengfei Jin, Mingze Yuan, Na Li, Bo Zeng, and Quanzheng Li. RODS: Robust optimization inspired diffusion sampling for detecting and reducing hallucination in generative models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=fhuqIxoPcr.

Matthew Tivnan, Siyeop Yoon, Zhennong Chen, Xiang Li, Dufan Wu, and Quanzheng Li. Hallucination index: An image quality metric for generative reconstruction models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 449–458. Springer, 2024.

Kostas Triaridis, Alexandros Graikos, Aggelina Chatziagapi, Grigorios G Chrysos, and Dimitris Samaras. Mitigating diffusion model hallucinations with dynamic guidance. *arXiv preprint arXiv:2510.05356*, 2025a.

Kostas Triaridis, Alexandros Graikos, Aggelina Chatziagapi, Grigorios G Chrysos, and Dimitris Samaras. Mitigating diffusion model hallucinations with dynamic guidance. *arXiv preprint arXiv:2510.05356*, 2025b.

Zhenzhen Weng, Laura Bravo-Sánchez, and Serena Yeung-Levy. Diffusion-hpc: Synthetic data generation for human mesh recovery in challenging domains. In *2024 International Conference on 3D Vision (3DV)*, pp. 257–267. IEEE, 2024.

Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7378–7387, 2023.

Dianbing Xi, Jiepeng Wang, Yuanzhi Liang, Xi Qiu, Yuchi Huo, Rui Wang, Chi Zhang, and Xuelong Li. Omnivdiff: Omni controllable video diffusion for generation and understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10915–10923, 2026.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. Unveiling the potential of diffusion large language model in controllable generation. In *International Conference on Learning Representations*, volume 2026, pp. 66670–66686, 2026.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14273–14282, 2026.

Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

Haozhuo Zhang, Bin Zhu, Yu Cao, and Yanbin Hao. Hand1000: Generating realistic hands from text with only 1,000 images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9905–9913, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Jie Zhu, Yixiong Chen, Mingyu Ding, Ping Luo, Leye Wang, and Jingdong Wang. Mole: Enhancing human-centric text-to-image diffusion via mixture of low-rank experts. *Advances in Neural Information Processing Systems*, 37:29354–29386, 2024.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

A Background

We first review the Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020). Given a data sample drawn from the real data distribution, $\boldsymbol{x}_0 \sim q(\boldsymbol{x})$, DDPM defines a *forward process* that gradually corrupts the data by adding Gaussian noise over a sequence of timesteps. This process is specified by the Markov transition: for $t \in \{1, \cdots, T\}$.

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}),$$

where $\beta_t \in (0, 1)$ denotes a predefined variance schedule controlling the noise magnitude at timestep t.

By applying the reparameterization trick, the forward process admits a closed-form expression for the marginal distribution of $\boldsymbol{x}_t$ conditioned on the original data point $\boldsymbol{x}_0$:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}),$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. This enables efficient training by allowing $\boldsymbol{x}_t$ to be sampled directly from $\boldsymbol{x}_0$ at any t without simulating the entire forward trajectory.

The generative procedure learns a *reverse process* that iteratively denoises a sample initialized from pure Gaussian noise $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. At each timestep, the model approximates the reverse conditional distribution $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, ultimately producing samples that follow $q(\boldsymbol{x})$. In particular, a noisy observation at timestep t can be written as:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_t,$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. A denoising diffusion model $\boldsymbol{\epsilon}_\theta$ is trained to predict the injected noise $\boldsymbol{\epsilon}_t$ from the noisy input $\boldsymbol{x}_t$ and timestep t, by minimizing the following objective:

$$\mathcal{L} = \mathbb{E}_{t\sim[1,T],\boldsymbol{x}_0,\boldsymbol{\epsilon}_t}\Big[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\|^2\Big],$$

where $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ denotes the model's noise prediction.

Sampling proceeds by iteratively applying the learned reverse diffusion dynamics. For each timestep $t \in \{T, \ldots, 1\}$, the reverse update is

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \sigma_t \boldsymbol{z},$$

where $\sigma_t = \sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}$, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ for $t > 1$, and $\boldsymbol{z} = \boldsymbol{0}$ at the final step. This stochastic update combines a deterministic denoising term with Gaussian noise at each timestep.

Moreover, DDPMs admit a simple approximation of the gradient of the log-density with respect to $\boldsymbol{x}_t$ in terms of the learned noise predictor. In particular, the (Stein's) score function can be written as

$$\boldsymbol{s}_\theta(\boldsymbol{x}_t) \overset{\text{def}}{=} \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t) \approx -\frac{\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}},$$

which establishes a direct connection between DDPMs and score-based models (Song & Ermon, 2019; Chan et al., 2024).

B Proof of Theorem 1

Consider a mixture of two Gaussian distributions, $\boldsymbol{x}_0 \sim \pi\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{I}) + (1-\pi)\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{I})$, where $\pi \in (0, 1)$, and noisy observations $\boldsymbol{x}_t$ are generated according to

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$$

The predicted clean image $\hat{\boldsymbol{x}}_0^{(t)}$ is computed by the posterior mean predictor (PMP)

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

Assume the trained $\boldsymbol{\epsilon}_\theta$ matches the ground-truth distribution, so that $\hat{\boldsymbol{x}}_0^{(t)} = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$. We prove that the leading eigenvector of $\mathrm{Cov}(\hat{\boldsymbol{x}}_0^{(t)})$ is parallel to $\boldsymbol{\delta} \coloneqq \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$.

Proof. We first derive $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t, k]$ and $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$. Define

$$\boldsymbol{y} \coloneqq \frac{\boldsymbol{x}_t}{\sqrt{\bar{\alpha}_t}} = \boldsymbol{x}_0 + \sigma\boldsymbol{\epsilon},\ \sigma^2 = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

By Tweedie's Formula (Efron, 2011; Lai et al., 2025), we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{y}, k] &= y + \sigma^2 \nabla_{\boldsymbol{y}} \log q(y \mid k) \\
&= y + \sigma^2 \nabla_{\boldsymbol{y}} \left(-\frac{\|\boldsymbol{y} - \boldsymbol{\mu}_k\|^2}{2(1+\sigma^2)} \right) \\
&= y + \sigma^2 \left(\frac{\boldsymbol{\mu}_k - y}{1+\sigma^2} \right) \\
&= \frac{1}{1+\sigma^2}\boldsymbol{y} + \frac{\sigma^2}{1+\sigma^2}\boldsymbol{\mu}_k.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t, k] &= \frac{1}{\sqrt{\bar{\alpha}_t}(1+\sigma^2)}\boldsymbol{x}_t + \frac{\sigma^2}{\sqrt{\bar{\alpha}_t}(1+\sigma^2)}\boldsymbol{\mu}_k \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{\mu}_k
\end{aligned}$$

Let $\mathcal{W}(\boldsymbol{x}_t) \coloneqq q(k=0 \mid \boldsymbol{x}_t)$, then

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] &= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_t + (1-\bar{\alpha}_t)(\mathcal{W}(\boldsymbol{x}_t)\boldsymbol{\mu_1} + (1 - \mathcal{W}(\boldsymbol{x}_t)\boldsymbol{\mu}_1)) \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_t + (1-\bar{\alpha}_t)(\mathcal{W}(\boldsymbol{x}_t)(\boldsymbol{\mu_2} + \boldsymbol{\delta}) + (1 - \mathcal{W}(\boldsymbol{x}_t))\boldsymbol{\mu}_1)) \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_t + (1-\bar{\alpha}_t)(\boldsymbol{\mu_2} + \mathcal{W}(\boldsymbol{x}_t)\boldsymbol{\delta}).
\end{aligned}$$

Since $\hat{\boldsymbol{x}}_0^{(t)} = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$, the law of total expectation gives

$$\mathbb{E}[\hat{\boldsymbol{x}}_0^{(t)}] = \mathbb{E}[\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]] = \mathbb{E}[\boldsymbol{x}_0].$$

Let $\boldsymbol{\mu} \coloneqq \mathbb{E}[\boldsymbol{x}_0]$, we have

$$\boldsymbol{\mu} = \pi\boldsymbol{\mu}_0 + (1-\pi)\boldsymbol{\mu}_1 = \pi(\boldsymbol{\mu}_1 + \boldsymbol{\delta}) + (1-\pi)\boldsymbol{\mu}_1 = \pi\boldsymbol{\delta} + \boldsymbol{\mu}_1.$$

We can now write

$$\begin{aligned}
\hat{\boldsymbol{x}}_0^{(t)} - \boldsymbol{\mu} &= \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] - \boldsymbol{\mu} \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_t + (1-\alpha_t)\boldsymbol{\mu}_1 + (1-\bar{\alpha}_t)\mathcal{W}(\boldsymbol{x}_t)\boldsymbol{\delta} - \boldsymbol{\mu} \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_t + (1-\alpha_t)(\boldsymbol{\mu} - \pi\boldsymbol{\delta}) + (1-\bar{\alpha}_t)\mathcal{W}(\boldsymbol{x}_t)\boldsymbol{\delta} - \boldsymbol{\mu} \\
&= \underbrace{\sqrt{\bar{\alpha}_t}(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu})}_{\boldsymbol{A}} + \underbrace{(1-\bar{\alpha}_t)(\mathcal{W}(\boldsymbol{x}_t) - \pi)\boldsymbol{\delta}}_{\boldsymbol{B}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathrm{Cov}(\hat{\boldsymbol{x}}_0^{(t)}) &= \mathbb{E}[(\boldsymbol{A}+\boldsymbol{B})(\boldsymbol{A}+\boldsymbol{B})^\top] \\
&= \mathbb{E}[\boldsymbol{A}\boldsymbol{A}^\top] + \mathbb{E}[\boldsymbol{A}\boldsymbol{B}^\top] + \mathbb{E}[\boldsymbol{B}\boldsymbol{A}^\top] + \mathbb{E}[\boldsymbol{B}\boldsymbol{B}^\top].
\end{aligned}$$

Since $\mathbb{E}[\boldsymbol{x}_t] = \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}$, we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{A}\boldsymbol{A}^\top] &= \bar{\alpha}_t \mathrm{Cov}(\boldsymbol{x}_t) \\
&= \bar{\alpha}_t \mathbb{E}[\mathrm{Cov}[\boldsymbol{x}_t \mid k]] + \bar{\alpha}_t \mathrm{Cov}[\mathbb{E}[\boldsymbol{x}_t \mid k]] \quad \text{(Law of total covariance)} \\
&= \bar{\alpha}_t \boldsymbol{I} + \bar{\alpha}_t^2 \mathrm{Cov}[\boldsymbol{\mu}_k] \\
&= \bar{\alpha}_t \boldsymbol{I} + \bar{\alpha}_t(\pi(\boldsymbol{\mu}_0 - \boldsymbol{\mu})(\boldsymbol{\mu}_0 - \boldsymbol{\mu})^\top + (1-\pi)(\boldsymbol{\mu}_1 - \boldsymbol{\mu})(\boldsymbol{\mu}_1 - \boldsymbol{\mu})) \\
&= \bar{\alpha}_t \boldsymbol{I} + \bar{\alpha}_t(\pi(1-\pi)^2 \boldsymbol{\delta}\boldsymbol{\delta}^\top + (1-\pi)\pi^2 \boldsymbol{\delta}\boldsymbol{\delta}^\top) \\
&= \bar{\alpha}_t \boldsymbol{I} + \bar{\alpha}_t^2 \pi(1-\pi)\boldsymbol{\delta}\boldsymbol{\delta}^\top.
\end{aligned}$$

According to the law of total expectation, we have

$$\mathbb{E}[\mathcal{W}(\boldsymbol{x}_t)] = \mathbb{E}[q(k=0 \mid \boldsymbol{x}_t)] = \mathbb{E}[\mathbb{E}[\mathbf{1}(k=0) \mid \boldsymbol{x}_t] = \mathbb{E}[\mathbf{1}(k=0)] = \pi,$$

where $\mathbf{1}(\cdot)$ is the indicator function, then we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{B}\boldsymbol{B}^\top] &= \mathbb{E}[(1-\bar{\alpha}_t)^2(\mathcal{W}(\boldsymbol{x}_t) - \pi)^2 \boldsymbol{\delta}\boldsymbol{\delta}^\top] \\
&= (1-\bar{\alpha}_t)^2 \mathbb{E}[(\mathcal{W}(\boldsymbol{x}_t) - \mathbb{E}[\mathcal{W}(\boldsymbol{x}_t)])^2] \boldsymbol{\delta}\boldsymbol{\delta}^\top \\
&= (1-\bar{\alpha}_t)^2 \mathrm{Var}[\mathcal{W}(\boldsymbol{x}_t)] \boldsymbol{\delta}\boldsymbol{\delta}^\top.
\end{aligned}$$

Using $\mathbb{E}[\boldsymbol{x}_t] = \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}$, $\mathbb{E}[\mathcal{W}(\boldsymbol{x}_t)] = \pi$, and $\boldsymbol{\mu}_0 - \boldsymbol{\mu} = (1-\pi)\boldsymbol{\delta}$, we obtain

$$\begin{aligned}
\mathbb{E}[\boldsymbol{A}\boldsymbol{B}^\top] &= \mathbb{E}[\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu})(\mathcal{W}(\boldsymbol{x}_t) - \pi)\boldsymbol{\delta}^\top] \\
&= \sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)\mathrm{Cov}[\boldsymbol{x}_t, \mathcal{W}(\boldsymbol{x}_t)]\boldsymbol{\delta}^\top \\
&= \sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)\Big(\mathbb{E}[\boldsymbol{x}_t \mathcal{W}(\boldsymbol{x}_t)] - \mathbb{E}[\boldsymbol{x}_t]\mathbb{E}[\mathcal{W}(\boldsymbol{x}_t)]\Big)\boldsymbol{\delta}^\top \\
&= \sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)(\pi\sqrt{\bar{\alpha}_t}\boldsymbol{\mu_1} - \sqrt{\bar{\alpha}_t}\pi\boldsymbol{\mu})\boldsymbol{\delta}^\top \\
&= \bar{\alpha}_t(1-\bar{\alpha}_t)\pi(1-\pi)\boldsymbol{\delta}\boldsymbol{\delta}^\top.
\end{aligned}$$

Similarly, we can prove $\mathbb{E}[\boldsymbol{B}\boldsymbol{A}^\top] = \bar{\alpha}_t(1-\bar{\alpha}_t)\pi(1-\pi)\boldsymbol{\delta}\boldsymbol{\delta}^\top$. Summing all terms yields

$$\mathrm{Cov}[\hat{\boldsymbol{x}}_0^{(t)}] = \bar{\alpha}_t \boldsymbol{I} + \underbrace{\Big((2\bar{\alpha}_t - \bar{\alpha}_t^2)\pi(1-\pi) + (1-\bar{\alpha}_t)^2 \mathrm{Var}[\mathcal{W}(\boldsymbol{x}_t)]\Big)}_{b} \boldsymbol{\delta}\boldsymbol{\delta}^\top,$$

where b is a scalar. For any vector $\boldsymbol{v}_1$ orthogonal to $\boldsymbol{\delta}$, we have

$$\mathrm{Cov}[\hat{\boldsymbol{x}}_0^{(t)}]\boldsymbol{v}_1 = \bar{\alpha}_t \boldsymbol{v}_1,$$

so $\boldsymbol{v}_1$ is en eigenvector with eigenvalue $\bar{\alpha}_t$. For any vector $\boldsymbol{v}_2 = c\boldsymbol{\delta}$ parallel to $\boldsymbol{\delta}$ with scalar c, we have

$$\mathrm{Cov}[\hat{\boldsymbol{x}}_0^{(t)}]\boldsymbol{v}_2 = \bar{\alpha}_t \boldsymbol{v}_2 + bc\|\boldsymbol{\delta}\|^2 \boldsymbol{\delta},$$

so v_2 is en eigenvector with eigenvalue $\bar{\alpha}_t + bc\|\delta\|^2$. When

$$b \coloneqq (2\bar{\alpha}_t - \bar{\alpha}_t^2)\pi(1-\pi) + (1-\bar{\alpha}_t)^2 \mathrm{Var}[\mathcal{W}(\boldsymbol{x}_t)] > 0,$$

$\boldsymbol{v}_2$ is an eigenvector with strictly larger eigenvalue, which always hold when $\bar{\alpha}_t \in (0,1)$ and $\pi \in (0,1)$.

□

C Proof of Theorem 2

Under the same Gaussian mixture setting as in Theorem 3.1, we show that steering $\boldsymbol{x}_t$ with vector $\Delta\boldsymbol{x} \approx \lambda\sqrt{\bar{\alpha}_t}\boldsymbol{\delta}_t$ is equivalent to applying classifier guidance.

Proof. Recall that under the Gaussian mixture model in Theorem 3.1, we have the following conditional distribution

$$q(\boldsymbol{x}_t \mid k) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{\mu}_k, \boldsymbol{I}),$$

where k is the latent class. In classifier guidance, the class-conditional score function can be decomposed as

$$\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t \mid k) = \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log q(k \mid \boldsymbol{x}_t).$$

We can derive the second term $\nabla_{\boldsymbol{x}_t} \log q(k \mid \boldsymbol{x}_t)$. Define $\mathcal{W}(\boldsymbol{x}_t) = q(k = 0 \mid \boldsymbol{x}_t)$, then $q(k = 1 \mid \boldsymbol{x}_t) = 1 - \mathcal{W}(\boldsymbol{x}_t)$. To compute its gradient, we define

$$\mathcal{S}(x) \coloneqq \frac{\exp x}{\exp x + 1}, \quad \mathcal{L}(x) = \log \frac{\mathcal{W}(x)}{1 - \mathcal{W}(x)},$$

then we have

$$\mathcal{W}(x) = \mathcal{S}(\mathcal{L}(x)).$$

Taking gradient gives

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t} \log \mathcal{W}(\boldsymbol{x}_t) &= \frac{1}{\mathcal{W}(\boldsymbol{x}_t)} \nabla_{\boldsymbol{x}_t} \mathcal{W}(\boldsymbol{x}_t) \\
&= \frac{1}{\mathcal{W}(\boldsymbol{x}_t)} \nabla_{\boldsymbol{x}_t} \mathcal{S}(\mathcal{L}(\boldsymbol{x}_t)) \\
&= \frac{1}{\mathcal{W}(\boldsymbol{x}_t)} \mathcal{S}(\mathcal{L}(\boldsymbol{x}_t))(1 - \mathcal{S}(\mathcal{L}(\boldsymbol{x}_t))) \nabla_{\boldsymbol{x}_t} \mathcal{L}(\boldsymbol{x}_t) \\
&= \frac{1}{\mathcal{W}(\boldsymbol{x}_t)} \mathcal{W}(\boldsymbol{x}_t)(1 - \mathcal{W}(\boldsymbol{x}_t)) \nabla_{\boldsymbol{x}_t} \mathcal{L}(\boldsymbol{x}_t) \\
&= (1 - \mathcal{W}(\boldsymbol{x}_t)) \nabla_{\boldsymbol{x}_t} \mathcal{L}(\boldsymbol{x}_t).
\end{aligned}$$

Analyze the log-odds under the Gaussian mixture model, we obtain

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_t) &= \log \frac{q(k = 0 \mid \boldsymbol{x}_t)}{q(k = 1 \mid \boldsymbol{x}_t)} = \log \frac{q(\boldsymbol{x}_t \mid k = 0) \cdot q(k = 0)}{q(\boldsymbol{x}_t \mid k = 1) \cdot q(k = 1)} \\
&= \log \frac{\pi}{1 - \pi} + \log q(\boldsymbol{x}_t \mid k = 0) - \log q(\boldsymbol{x}_t \mid k = 1) \\
&= \log \frac{\pi}{1 - \pi} - \frac{1}{2}(\|\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}_0\|^2 - \|\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}_1\|^2) \\
&= \log \frac{\pi}{1 - \pi} - \frac{1}{2}\Big(- 2\sqrt{\bar{\alpha}_t}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top \boldsymbol{x}_t + \bar{\alpha}_t(\|\boldsymbol{\mu}_0\|^2 - \|\boldsymbol{\mu}_1\|^2)\Big) \\
&= \log \frac{\pi}{1 - \pi} + \sqrt{\bar{\alpha}_t}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top \boldsymbol{x}_t - \frac{\bar{\alpha}_t}{2}(\|\boldsymbol{\mu}_0\|^2 - \|\boldsymbol{\mu}_1\|^2).
\end{aligned}$$

Hence we have: $\nabla_{\boldsymbol{x}_t} \mathcal{L}(\boldsymbol{x}_t) = \sqrt{\bar{\alpha}_t}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) = \sqrt{\bar{\alpha}_t}\boldsymbol{\delta}$. When $k = 0$, we have

$$\nabla_{\boldsymbol{x}_t} \log q(k = 0 \mid \boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log \mathcal{W}(\boldsymbol{x}_t) = \sqrt{\bar{\alpha}_t}(1 - \mathcal{W}(\boldsymbol{x}_t))\boldsymbol{\delta}.$$

Similarly, when $k = 1$, we have

$$\nabla_{\boldsymbol{x}_t} \log q(k = 1 \mid \boldsymbol{x}_t) = -\sqrt{\bar{\alpha}_t}\mathcal{W}(\boldsymbol{x}_t)\boldsymbol{\delta}.$$

Define $\mathcal{P}(\boldsymbol{x}_t, k) \coloneqq q(k \mid \boldsymbol{x}_t)$, $\boldsymbol{\delta}^{(k)} \coloneqq \boldsymbol{\mu}_k - \boldsymbol{\mu}_{1-k}$, then we know $\nabla_{\boldsymbol{x}_t} \log q(k \mid \boldsymbol{x}_t) = \sqrt{\bar{\alpha}_t}(1 - \mathcal{P}(\boldsymbol{x}_t, k))\boldsymbol{\delta}^{(k)}$. Now we have the score function under classifier guidance with strengthen w

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t \mid k) &= \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t) + w \nabla_{\boldsymbol{x}_t} \log q(k \mid \boldsymbol{x}_t) \\
&= \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t) + w\sqrt{\bar{\alpha}_t}(1 - \mathcal{P}(\boldsymbol{x}_t))\boldsymbol{\delta}^{(k)}.
\end{aligned}$$

Using the standard correspondence between the score function and the noise predictor, we derive the modified noise prediction under classifier guidance. Given $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t) \approx -\frac{\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{1 - \bar{\alpha}_t}}$, then we can replace the original noise predictor $\boldsymbol{\epsilon}_\theta$.

$$\begin{aligned}
-\frac{\bar{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{1 - \bar{\alpha}_t}} &= -\frac{\epsilon_\theta(\boldsymbol{x}_t, t)}{\sqrt{1 - \bar{\alpha}_t}} + w\sqrt{\bar{\alpha}_t}(1 - \mathcal{P}(\boldsymbol{x}_t))\boldsymbol{\delta}^{(k)} \\
\bar{\epsilon}_\theta(\boldsymbol{x}_t, t) &= \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - w\sqrt{\bar{\alpha}_t(1 - \bar{\alpha}_t)}(1 - \mathcal{P}(\boldsymbol{x}_t))\boldsymbol{\delta}^{(k)}
\end{aligned}$$

Recall the DDIM sampling steps

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

we can relate perturbation on $\boldsymbol{x}_t$ to perturbation on $\boldsymbol{\epsilon}_\theta$. Let

$$\begin{aligned} &\sqrt{\bar{\alpha}_{t-1}} \left(\frac{(\boldsymbol{x}_t + \Delta\boldsymbol{x}) - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t + \Delta\boldsymbol{x}, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\, \boldsymbol{\epsilon}_\theta((\boldsymbol{x}_t + \Delta\boldsymbol{x}), t) \\ &= \sqrt{\bar{\alpha}_{t-1}} \left(\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \Delta\epsilon)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\, (\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \Delta\epsilon), \end{aligned}$$

and assume $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \approx \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t + \Delta\boldsymbol{x}, t)$, we have

$$\Delta\boldsymbol{x} = \left(\sqrt{\frac{\bar{\alpha}_t(1-\bar{\alpha}_{t-1})}{\bar{\alpha}_{t-1}}} - \sqrt{1-\bar{\alpha}_t} \right) \Delta\boldsymbol{\epsilon}.$$

Given

$$\Delta\boldsymbol{x} = \lambda\sqrt{\bar{\alpha}_t}\boldsymbol{\delta}, \quad \Delta\boldsymbol{\epsilon} = w\sqrt{\bar{\alpha}_t(1-\bar{\alpha}_t)}(1 - \mathcal{P}(\boldsymbol{x}_t, k))\boldsymbol{\delta}^{(k)},$$

we have

$$w = \pm \frac{\lambda\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t(1-\bar{\alpha}_t)(1-\bar{\alpha}_{t-1})} - \sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)(1-\mathcal{P}(\boldsymbol{x}_t, k))}}.$$

□

D Experimental Details

D.1 Datasets and Models

(i) MNIST (LeCun et al., 2010). The MNIST dataset consists of 60,000 grayscale handwritten digit images from 10 classes, each with resolution 28×28. The dataset is distributed under the Creative Commons Attribution-Share Alike 3.0 license. We train a DDPM model on the training set and sample 100,000 synthetic images. To quantify hallucination and validity, we use a pretrained classifier that achieves 99% accuracy on MNIST to compute Inception Score (IS) (Salimans et al., 2016). We select 10,000 images with the smallest IS as hallucinated samples, interpreting low image quality as indicative of hallucinated digit structures. We measure *validity* using IS. We measure *fidelity* using LPIPS (Zhang et al., 2018), which quantifies perceptual similarity between the original and edited images.

(ii) Simple Shapes (Aithal et al., 2024). The Simple Shapes dataset contains grayscale images of resolution 64×64, each depicting shapes including triangle, square, and pentagon. We generate this dataset following the rules proposed in Aithal et al. (2024). By construction, no image contains more than one instance of any shape. Using a DDPM trained on this dataset, we sample 100,000 synthetic images, of which 14,094 are identified as hallucinated. We automatically detect shape categories and their occurrences in each synthetic image using OpenCV. An image is labeled as hallucinated if any shape appears at least twice, or no shape exists. For evaluation, an edited image is considered *valid* if at least one shape exists, and no shape occurs more than once. It satisfies *fidelity* if it contains the same types of shapes as the original image, without missing or introducing additional shapes. This metric is more sensitive than LPIPS.

(iii) Hands (Afifi, 2019). The Hands dataset consists of 11,076 photographs of human hands against a uniform white background, resized to 128×128. The images include variations in hand orientation (left or right), pose (open or closed fingers), and viewpoint (dorsal or palmar). The dataset is free for reasonable academic fair use. We train a DDPM on this dataset and sample 10,000 synthetic images. From these, we use MediaPipe Hands (Zhang et al., 2020) to automatically detect 2,000 hallucinated samples with the lowest hand-detection confidence score returned by the detector. *Validity* is measured by MediaPipe's detection confidence on the edited image. *Fidelity* is measured using LPIPS.

(iv) Shaking Hands. We evaluate our method on a text-to-image diffusion model DeepFloyd-IF (DeepFloyd), which first generates images at resolution 64×64 and subsequently upsamples

them to 256×256. We use the prompt *"Shaking hands"* to generate 10,000 synthetic images. This prompt has also been adopted in prior work to illustrate the limitations of text-to-image diffusion models when generating concepts that are underrepresented in the training data (Samuel et al., 2024). Similar to Hands, we use MediaPipe Hands to select 1,000 hallucinated samples with the lowest confidence score. *Validity* is measured by MediaPipe's detection confidence, and *fidelity* is measured using LPIPS.

All DDPM models are implemented using the UNet2DModel from the HuggingFace Diffusers library, trained with 1,000 diffusion timesteps and sampled with 500 inference steps. For the Shaking Hands dataset, we use the DeepFloyd-IF models IF-I-M-v1.0 (stage 1) and IF-II-M-v1.0 (stage 2), sampling with 30 inference steps and a guidance strength of 7.0.

D.2 Baselines

We compare our *UTS* algorithm against three baselines:

(i) DDIM Inversion (DI) (Song et al., 2021). This baseline reverses an image to a noisy latent via DDIM inversion and then reconstructs the image using standard DDIM sampling. The reconstructed image typically exhibits only subtle differences from the original. In some cases, DDIM Inversion alone can partially alleviate hallucinations. This baseline can be viewed as a special case of our method without applying any steering, serving to isolate the effect of image reconstruction. The hyperparameters are the number of inference steps T and the reversed timestep $\bar{T}$.

(ii) Randomized Trajectory Steering (RTS). This variant replaces the steering direction in our algorithm with a random vector of the same norm at each timestep. By comparing RTS with UTS, we assess the importance of using the principal component, as opposed to arbitrary perturbations, for hallucination correction. RTS uses the same set of hyperparameters as UTS, including the number of inference steps T, the starting timestep $\bar{T}$, the ending timestep $\bar{t}$, and the perturbation strength λ.

(iii) CAS (Tian et al., 2025). CAS mitigates hallucinations by identifying high-risk sampling steps and intervening in the update rule through a robust optimization procedure. Unlike our approach, CAS relies on the local curvature of the loss landscape at inference time and does not exploit the uncertainty of PMP inherent in the diffusion model's denoising dynamics. The hyperparameters are the detection threshold h based on score magnitude fluctuations, a detection radius ρ, and the number of perturbed steps τ. We tune the detection threshold to ensure that at least 50% of annotated hallucinated images are successfully detected.

(iv) Dynamic Guidance (DG) (Triaridis et al., 2025a). Motivated by Aithal et al. (2024), Dynamic Guidance (DG) adaptively sharpens the learned score function during sampling to mitigate hallucinations caused by over-smoothing. DG first identifies a set of mode labels, then during sampling alternately infers the most probable mode given the current noisy latent and applies guidance toward that mode. Unlike UTS, DG relies on external supervision to define and identify modes; we nonetheless include it as a baseline for comparison. Hyperparameters of DG include the number of inference steps T, the starting timestep $\bar{T}$, the ending timestep $\bar{t}$, and the perturbation strength λ.

D.3 Hyperparameters

The hyperparameters used to compute Table 1 are summarized in Table 2. Hyperparameter settings that achieve the highest validity are underlined. Qualitative results corresponding to these settings are shown in Fig. 4 and 6–9. The ablation study in Fig. 5a is conducted with $T = 32$ and $\lambda = 10$. Fig. 5b is evaluated with $T = 32$, $\bar{T} = 26$, and $\bar{t} = 0$. Fig. 5c is evaluated with $T = 32$, $\bar{T} = 26$, $\bar{t} = 0$, and $\lambda = 10$.

D.4 Qualitative Results

Additional qualitative results are presented in Fig. 6–9. While DI and RTS occasionally mitigate hallucinations, they often reproduce hallucinated artifacts or introduce additional blurriness. CAS is less effective than UTS at correcting hallucinations. We attribute this to the fact that UTS leverages PMP, which encodes rich semantic information for targeted correction.

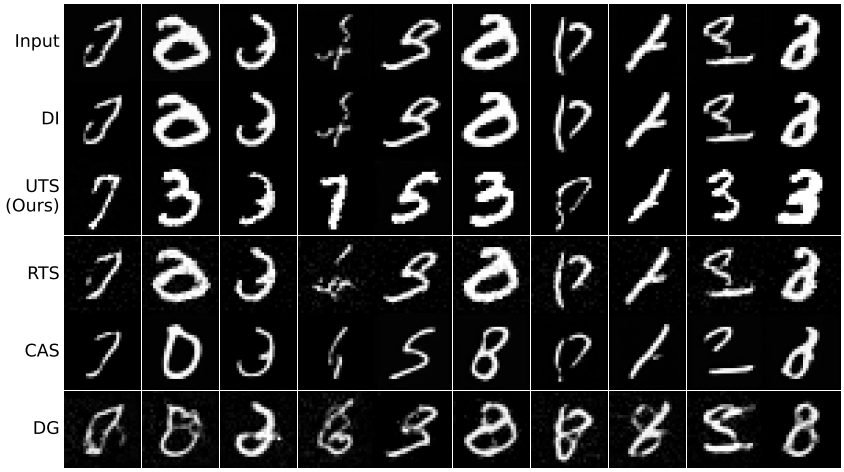


Figure 6: Qualitative results on the MNIST dataset.

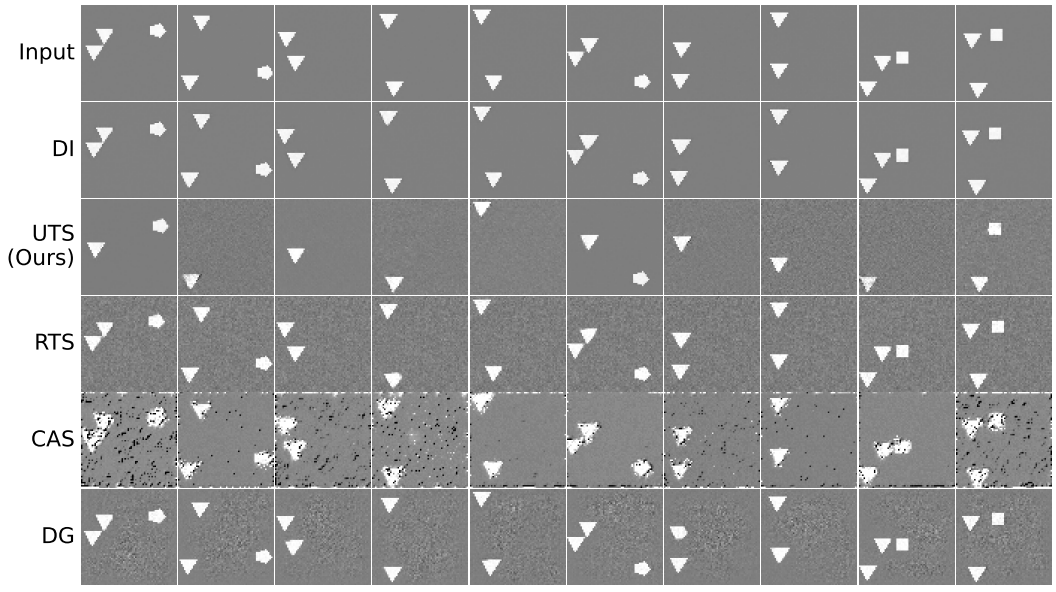


Figure 7: Qualitative results on the Simple Shapes dataset.

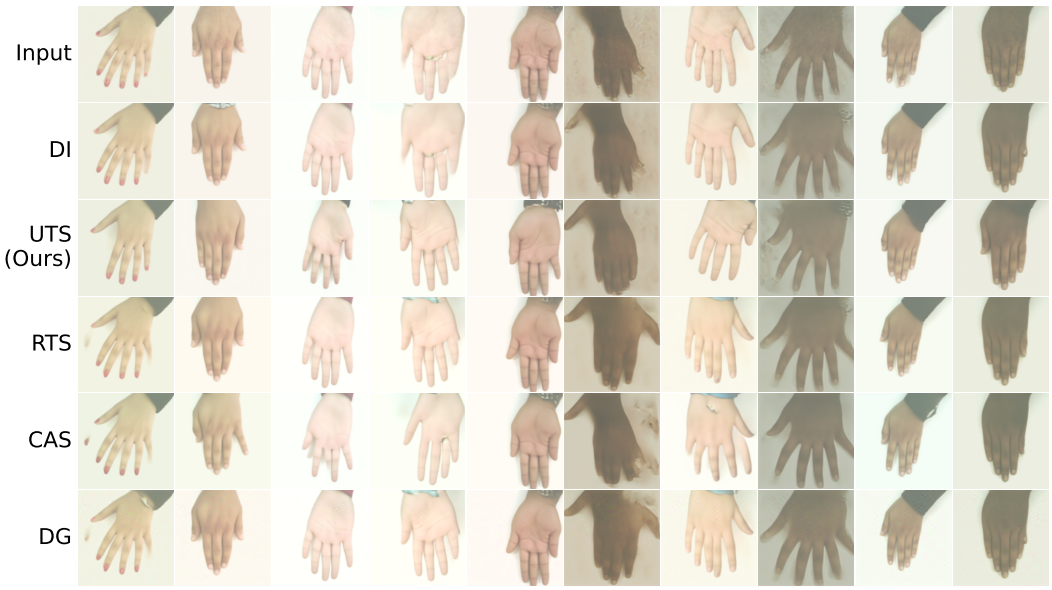


Figure 8: Qualitative results on the Hands dataset.



Figure 9: Qualitative results on the Shaking Hands dataset.

Method	Hyperparameter	MNIST	Simple Shapes	Hands	Shaking Hands
DI ($T = 16$)	Start Timestep $\bar{T}$	$\{13, 15\}$	$\{13, 15\}$	$\{13, 15\}$	–
DI ($T = 32$)	Start Timestep $\bar{T}$	$\{\underline{26}, 31\}$	$\{\underline{26}, 31\}$	$\{\underline{26}, 31\}$	$\{\underline{20}, 25\}$
DI ($T = 64$)	Start Timestep $\bar{T}$	–	–	–	$\{40, 50\}$
UTS ($T = 16$)	Start Timestep $\bar{T}$	$\{13, 15\}$	$\{13, 15\}$	$\{13, 15\}$	–
	End Timestep $\bar{t}$	0	0	5	–
	Steering strengthen λ	$\{5, 10\}$	$\{10, 15\}$	$\{5, 10\}$	–
	# PCA samples	10	10	10	–
UTS ($T = 32$)	Start Timestep $\bar{T}$	$\{\underline{26}, 31\}$	$\{\underline{26}, 31\}$	$\{\underline{26}, 31\}$	$\{20, 25\}$
	End Timestep $\bar{t}$	0	0	10	10
	Steering strengthen λ	$\{\underline{5}, 10\}$	$\{\underline{10}, 15\}$	$\{\underline{5}, 10\}$	$\{5, 10\}$
	# PCA samples	10	10	10	20
UTS ($T = 64$)	Start Timestep $\bar{T}$	–	–	–	$\{\underline{40}, 50\}$
	End Timestep $\bar{t}$	–	–	–	20
	Steering strengthen λ	–	–	–	$\{\underline{5}, 10\}$
	# PCA samples	–	–	–	20
RTS ($T = 16$)	Start Timestep $\bar{T}$	$\{13, 15\}$	$\{13, 15\}$	$\{13, 15\}$	–
	End Timestep $\bar{t}$	0	0	10	–
	Steering strengthen λ	$\{5, 10\}$	$\{10, 15\}$	$\{5, 10\}$	–
RTS ($T = 32$)	Start Timestep $\bar{T}$	$\{\underline{26}, 31\}$	$\{\underline{26}, 31\}$	$\{26, \underline{31}\}$	$\{20, 25\}$
	End Timestep $\bar{t}$	0	0	10	10
	Steering strengthen λ	$\{\underline{5}, 10\}$	$\{\underline{10}, 15\}$	$\{5, \underline{10}\}$	$\{5, 10\}$
RTS ($T = 64$)	Start Timestep $\bar{T}$	–	–	–	$\{\underline{40}, 50\}$
	End Timestep $\bar{t}$	–	–	–	20
	Steering strengthen λ	–	–	–	$\{\underline{5}, 10\}$
CAS ($T = 16$)	Detection threshold h	$\{0.01, 0.03\}$	$\{0.005, 0.01\}$	$\{0.002, 0.005\}$	–
	Detection radius ρ	$\{1, 2, 5\}$	$\{10, 20, 50\}$	$\{15, 30, 45\}$	–
	# Perturbed Steps τ	1	1	1	–
CAS ($T = 32$)	Detection threshold h	$\{0.01, \underline{0.03}\}$	$\{\underline{0.005}, 0.01\}$	$\{\underline{0.002}, 0.005\}$	$\{0.01, 0.02\}$
	Detection radius ρ	$\{1, 2, \underline{5}\}$	$\{10, 20, \underline{50}\}$	$\{15, \underline{30}, 45\}$	$\{0.2, 0.5, 1.0\}$
	# Perturbed Steps τ	1	1	1	1
CAS ($T = 64$)	Detection threshold h	–	–	–	$\{0.01, \underline{0.02}\}$
	Detection radius ρ	–	–	–	$\{0.2, 0.5, \underline{1.0}\}$
	# Perturbed Steps τ	–	–	–	1
DG ($T = 16$)	Start Timestep $\bar{T}$	$\{13, \underline{15}\}$	$\{13, 15\}$	$\{13, 15\}$	–
	End Timestep $\bar{t}$	0	0	0	–
	Steering strengthen λ	$\{\underline{5}, 10\}$	$\{10, 15\}$	$\{5, 10\}$	–
DG ($T = 32$)	Start Timestep $\bar{T}$	$\{26, 31\}$	$\{26, \underline{31}\}$	$\{26, \underline{31}\}$	$\{20, 25\}$
	End Timestep $\bar{t}$	0	0	0	0
	Steering strengthen λ	$\{5, 10\}$	$\{\underline{10}, 15\}$	$\{\underline{5}, 10\}$	$\{5, 10\}$
DG ($T = 64$)	Start Timestep $\bar{T}$	–	–	–	$\{40, \underline{50}\}$
	End Timestep $\bar{t}$	–	–	–	0
	Steering strengthen λ	–	–	–	$\{5, \underline{10}\}$

Table 2: Hyperparameters used for computing the validity–fidelity trade-off and visualizing qualitative results.

D.5 Inference Time

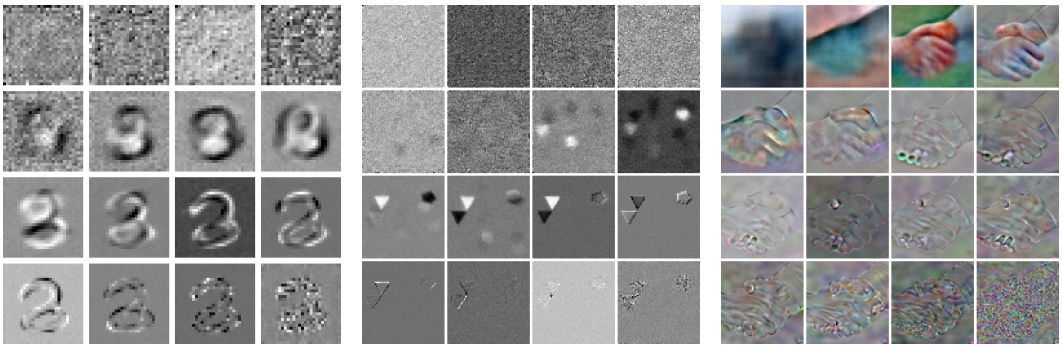
Table 3 reports the inference time (seconds per image) of UTS and the baselines (DI, CAS, DG) with $T = 32, \lambda = 10, \bar{T} = 26, \bar{t} = 5$, which can already achieve good performance in our experiments. The results show that UTS achieves inference time comparable to CAS and DG across all datasets, including on the larger text-to-image diffusion model used for Shaking Hands. The overhead relative to DI is expected, as DI performs a single forward pass at each step, whereas UTS and CAS both require multiple passes for uncertainty estimation.

Dataset	DI	CAS	DG	UTS (Ours)
MNIST	1.8	5.48	5.4	5.93
Simple Shapes	3.91	8.66	6.8	9.35
Hands	3.63	5.48	6.15	8.40
Shaking Hands	11.08	33.41	27.9	34.17

Table 3: Inference time (seconds per image) of UTS and baselines. UTS achieves comparable inference time to CAS and DG across all datasets. The overhead relative to DI is expected, as both UTS and CAS require extra passes for uncertainty estimation.

Should further acceleration be desired, the computational cost of SVD and multiple forward passes can be reduced by leveraging recent advances in denoiser uncertainty quantification (Manor & Michaeli, 2023). Specifically, Manor & Michaeli (2023) proposes a subspace iteration method to compute the leading principal components of the posterior covariance via efficient linear approximation of the Jacobian-vector product of the denoiser, avoiding explicit SVD entirely. The number of iterations required for convergence is small, further reducing the number of forward passes needed.

D.6 Principal Components

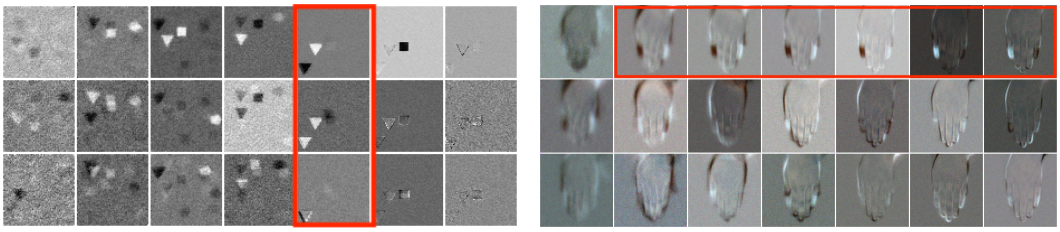


(a) MNIST (b) Simple Shapes (c) Shaking Hands

Figure 10: Trajectories of the first principal components along the denoising process. The upper-left panel corresponds to $t = 16$, and the sequence progresses to the bottom-right panel at $t = 1$. Uncertainty in the PMP at intermediate timesteps correlates with hallucinated artifacts, such as unrealistic digit strokes, redundant shapes, and extra fingers.

Fig. 10 visualizes the first principal components computed from different datasets given a hallucinated input image. The results indicate that uncertainty in the PMP at intermediate timesteps correlates with hallucinated artifacts, such as unrealistic digit strokes, redundant shapes, and extra fingers, which can be used to remove hallucinated structures from the original image.

We also visualize the higher-order principal components across multiple timesteps on Simple Shapes and Hands. As illustrated in Fig. 11, these components generally capture less reliable structure for hallucination removal compared to the leading component. In Simple Shapes, the leading principal component captures a semantically meaningful pattern of preserving one shape (white area) while suppressing a duplicate (black area), directly targeting hallucination. The second component may add or remove two shapes of the same type simultaneously, potentially increasing hallucination risk. The third component is blurrier and less structured. In Hands, the leading principal component consistently captures a finger-level correction pattern (lengthening one finger while shortening another), which is directly relevant to hallucination removal. Higher-order components capture this pattern less frequently and less precisely. These observations confirm that the leading principal component is the most semantically meaningful and reliable direction for steering.



(a) Simple Shapes (b) Hands

Figure 11: Visualization of the top-3 principal components at different timesteps (each column corresponds to one timestep, and each row corresponds to one principal component, with the first row being the leading component). These observations support the use of the leading principal component as the primary steering direction. In Simple Shapes, the second principal component tends to simultaneously add shapes of the same type, which may increase hallucination risk. In Hands, the higher-order components capture finger-level corrections less consistently.

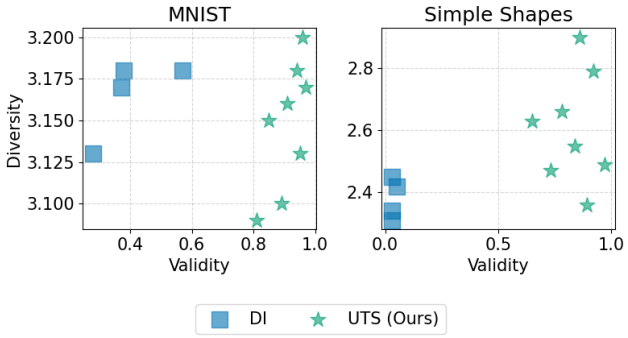


Figure 12: Diversity (label entropy) vs. validity across different hyperparameter settings on MNIST and Simple Shapes, where each point corresponds to one hyperparameter configuration. Results show that higher validity does not harm diversity, and our method can achieve higher diversity than DDIM Inversion.

D.7 Diversity-Validity Relationship

We empirically analyze the diversity-validity relationship on two datasets where class labels can be automatically obtained. For MNIST, we use a pretrained classifier to assign one of 10 digit classes. For Simple Shapes, diversity is measured over 8 classes defined by the presence or absence of each of the 3 shapes. We measure diversity as the entropy of the class distribution of steered images across different hyperparameter settings, with results visualized in Fig. 12. The results show that higher validity does not come at the expense of diversity. Our method occasionally achieves higher diversity than DDIM Inversion. We attribute this to the design of our method: since principal components are computed on a per-image basis rather than over a batch, and steering is applied dynamically along the denoising trajectory, our method performs a local exploration around each individual sample rather than collapsing toward global training modes. Consequently, the diversity of the edited outputs remains closely tied to the diversity of the original hallucinated images.

D.8 Hallucination Prevention

Our method can be applied to hallucination prevention during sampling by combining it with hallucination detection methods (Aithal et al., 2024; Tian et al., 2025). To demonstrate this, we evaluate UTS in the prevention setting: we sample new images from scratch, detect hallucinations during sampling, and steer the trajectory on the fly. We test on Simple Shapes, where hallucinations can be reliably and automatically detected. We sample 1,000 images. At each sampling step, we use OpenCV to detect hallucinated patterns in the predicted clean images, and apply a one-step steering update whenever a hallucination is detected. We also evaluate across two additional samplers beyond

DDIM. We report the hallucination rate and diversity (entropy over shape-class labels) in Table 4. Results show that our algorithm substantially reduces the hallucination rate across all three samplers, at a moderate cost in diversity.

Table 4: Hallucination rate and diversity across samplers on Simple Shapes, with and without UTS steering applied during sampling.

Sampler	Hallucination rate	Diversity
DDIM	0.123	2.7416
DDIM + UTS	**0.000**	2.1950
Euler-ancestral	0.152	2.6699
Euler-ancestral + UTS	**0.116**	2.6230
DPM	0.266	2.7451
DPM + UTS	**0.008**	2.2100

E Related Works

Hallucinations in Image Synthesis. Prior work has adopted diverse definitions of hallucinations in image synthesis, including malformed body structures (Lu et al., 2024), semantically meaningless text (Lu et al., 2025), and structural inaccuracies relative to ground truth (Kim et al., 2025). In this work, we primarily follow the definition proposed by Aithal et al. (2024), which characterizes hallucinations as samples lying outside the support of the training distribution. Under this definition, the notion of support may be specified either by explicit rules (e.g., the Simple Shapes dataset) or by implicit real-world distributions (e.g., the Hands dataset).

Aithal et al. (2024) hypothesize that such hallucinations arise from mode interpolation: diffusion models learn smooth approximations of the data distribution and may assign non-negligible probability mass to low-density regions between nearby high-density modes, leading to unrealistic samples. To detect hallucinations, they measure temporal variance of the posterior mean predictor (PMP) across adjacent timesteps, arguing that hallucinated trajectories exhibit increased instability. Subsequent work has explored alternative uncertainty signals. RODS (Tian et al., 2025) leverages spatial variance by estimating local curvature during sampling and modifying the update rule through robust optimization to enforce smoother trajectories. In contrast, our approach focuses on the inherent spatial variance of PMP conditioned on the timestep, which isolates uncertainty arising from mode ambiguity while disentangling it from the inherent coarse-to-fine denoising dynamics. Moreover, PMP encodes rich semantic information beyond local curvature, and has been shown to be effective for image editing (Chen et al., 2024), making it particularly suitable for targeted hallucination correction.

External Supervision for Hallucination Reduction. A large body of work mitigates hallucinations by incorporating external knowledge or auxiliary models. Triaridis et al. (2025b) relies on external labeling information to sharpen the score function during generation to mitigate hallucinations. HandRefiner (Lu et al., 2024) employs hand mesh reconstruction and image inpainting to correct malformed hand structures while preserving other regions. Diffusion-HPC (Weng et al., 2024) integrates human mesh recovery models, pose priors, and image segmentation to guide generation toward realistic human poses. Samuel et al. (2024) observe that text-to-image diffusion models struggle with rare concepts in imbalanced datasets, a phenomenon closely related to hallucination, and propose SeedSelect, which uses reference images to optimize initial noise seeds.

In contrast to these approaches, our method relies solely on the inherent knowledge encoded in a pretrained diffusion model, without requiring external supervision or additional training. While incorporating external knowledge may yield stronger corrections in certain settings, our results highlight the potential of diffusion models to self-correct hallucinations through inference-time interventions, and provide a foundation for future work that integrates internal and external signals more effectively.

Hallucinations in image reconstruction. Hallucinations have also been studied in image-conditioned diffusion models, particularly in medical imaging, where hallucinated structures in reconstructed images can lead to severe consequences such as incorrect diagnoses (Kim et al., 2024; 2025). Tivnan et al. (2024) propose a Hallucination Index to quantify deviations between reconstructed images and

a reference distribution with zero hallucination. Kim et al. (2024) focus on structural hallucination, defined as discrepancies between reconstructed images and ground truth distribution. They attribute such errors to out-of-distribution regions in the conditioning input, and mitigate hallucinations by separately generating outputs with in-distribution and out-of-distribution regions.

While our work primarily targets unconditional and text-to-image diffusion models, the underlying principle of steering sampling trajectories away from low-support regions suggests that our approach may also be applicable to conditional reconstruction tasks. We leave this extension to future work.

F LIMITATIONS

Our method is developed in the context of DDPM and DDIM diffusion models. Extending the idea of trajectory steering toward nearby modes to diffusion models formulated as differential equations remains an interesting direction for future work. In addition, the current implementation requires sampling multiple latents and computing principal components at inference time, which increases computational cost. Improving the efficiency of steering direction estimation is another promising avenue for further investigation.