Beyond Single-Point Perturbation: A Hierarchical, Manifold-Aware Approach to Diffusion Attacks

Anonymous submission

Abstract

Latent Diffusion Models have become a powerful tool for generating high-fidelity unrestricted adversarial examples. However, the existing methods typically perturb only the initial latent or rely on prompt engineering, which is ill-suited to the iterative nature of the diffusion process, plus optimization instability due to external text prompts and cumulative drift that push the adversarial images off the data manifold. In this paper, we propose a hierarchical attack framework that operates in alignment with the model's generative manifold and leverages intermediate denoising states to maximize attack transferability and visual fidelity. Extensive experiments show that the proposed attack improves adversarial transferability by 10-20% against a diverse set of normally-trained models and achieves over 10.5% higher success rate against adversarially-defended models, while simultaneously enhancing visual quality by 1.0-1.2 FID reduction and 16.7% LPIPS improvements.

Introduction

Deep Neural Networks have known vulnerabilities to adversarial examples that carefully crafted perturbations can induce erroneous predictions (Szegedy et al. 2013). While traditional L_p-constrained attacks often introduce perceptible, high-frequency noise (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017), *Unrestricted Adversarial Examples* have emerged as a new paradigm that generates natural-looking yet malicious examples through semantic manipulations of shape, color or textures (Xiao et al. 2018; Bhattad et al. 2019; Duan et al. 2021). The frontier for creating such high-fidelity adversaries lies within the framework of generative models, particularly the Latent Diffusion Models (LDMs) (Rombach et al. 2022), whose unparalleled ability to model complex distribution of natural images provides a powerful foundation to re-purpose them for adversarial attacks (Chen et al. 2023, 2024; Liu et al. 2023).

Unfortunately, inheriting from the earlier GAN-based attacks (Song et al. 2018), current methods are fundamentally constrained as they perform the entire adversarial optimization on a single, fixed timestep latent $\mathbf{z}_t$. This "single-point" approach finds an optimal "initial push" and then lets the generation process unfold in a "free-fall" of the denoising process without further course correction. Conceptually, this static paradigm creates a critical mismatch with the dynamic, iterative nature of diffusion, which progressively synthesizes an image by building high-level semantics in early stages and refining low-level details later (Yang et al. 2023).

Such mismatch further manifests in two critical instabilities of adversarial guidance and internal drift. First, existing attacks rely on Classifier-Free Guidance (CFG) (Ho and Salimans 2022), which uses external text prompts to steer the generation process. However, ambiguous or conflicting text prompts can introduce external noise and destabilize the generation trajectory by pushing the trajectory off the data manifold and causing visual artifacts (Sarıyıldız et al. 2023). Further, during the subsequent "free-fall" where denoising reconstruction proceeds without further correction, small errors from the initial push can accumulate at each step. This causes the adversarial path to progressively deviate from the valid manifold and degrades the final image quality and attack efficacy (Liu et al. 2022).

To overcome these limitations, we propose HAM (Hierarchical Attacks with Manifold Awareness on Diffusion Models), a holistic hierarchical framework that evolves beyond the existing single-point paradigms. However, the multi-step attack may exacerbate the challenge of error accumulation, where small, compounding drifts at each step would result in large deviations, especially with external text prompts. To tackle this challenge, our framework integrates manifold-awareness, and prompt-free internal guidance in a unified pipeline. First, in contrast to CFG with external noise, we propose a stable guidance via unconditional self-attention, which operates in a prompt-free mode to generate internally coherent guidance signals. Second, to counteract drifts amid multi-steps, we introduce dynamic manifold alignment, a regularization that continuously anchors the adversarial trajectory to the clean data statistics. Finally, these are further integrated into a manifold-constrained optimization that confines searches to the data manifold's tangent space. The main contributions are summarized below:

- ✧ Motivated by empirical studies, we shift from the static, single-point paradigm to a dynamic, hierarchical optimization in diffusion-based attacks that sequentially refines the adversarial latent throughout the entire generation process.
- ✧ We introduce a novel synergistic framework to tackle the cumulative error challenge in hierarchical attack. It uniquely combines prompt-free internal guidance and dynamic manifold alignment to ensure generation remains on the data manifold.
- ✧ Extensive experiments show that the proposed attack improves adversarial transferability by 10-20% against a diverse set of normally-trained models and achieves over

10.5% higher success rate against adversarially-defended models, while simultaneously enhancing visual quality by 1.0-1.2 FID reduction and 16.7% LPIPS improvements.

Related Work

Traditional Adversarial Attacks

Traditional adversarial attacks introduce imperceptible perturbations to images under strict L_p-norm constraints to ensure visual similarity. The seminal white-box attacks include gradient-based approaches like FGSM (Goodfellow, Shlens, and Szegedy 2014), PGD (Madry et al. 2017) (L_∞-norm), C&W (Carlini and Wagner 2017) and DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) (L_2-norm). While existing methods leverage data augmentation (Xie et al. 2019; Dong et al. 2019; Long et al. 2022), feature-space optimization (Inkawhich et al. 2019; Lu et al. 2020) and randomized smoothing (Xiao and Wang 2021) for enhancing black-box transferability, they inevitably introduce noticeable noise artifacts that compromise image naturalness. In contrast, our work, alongside other diffusion-based attacks (Chen et al. 2023, 2024), sidesteps this limitation by operating in the semantic latent space rather than pixel space.

Unrestricted Adversarial Attacks

Recognizing the limitations of L_p-norm for capturing perceptual distance, unrestricted adversarial attacks generate semantically meaningful perturbations without strict pixel-level constraints. Early methods manipulate specific attributes using GANs (Song et al. 2018; Bhattad et al. 2019; Jia et al. 2022; Qiu et al. 2020), apply spatial transformations (Xiao et al. 2018), or alter global color distributions (Yuan et al. 2022). While producing more natural examples, they often suffer from limited transferability. Powered by the success of Latent Diffusion Models (Rombach et al. 2022), diffusion-based attacks produce high-fidelity adversaries with superior transferability (Chen et al. 2024, 2023; Liu et al. 2023). However, the existing diffusion-based attacks have not investigated the synergistic benefits of jointly optimizing across multiple denoising stages and dynamically re-aligning perturbations with the latent data manifold. Moreover, prior attacks either do not generalize well in black-box settings or forgo perceptual quality. Our method uniquely addresses these open gaps through hierarchical, geometry-conscious multi-point optimization.

In addition, self-attention guidance in diffusion editing (Tumanyan et al. 2023) and manifold alignment via AdaIN (Huang and Belongie 2017) have been used for style transfer and image manipulation. However, they are not designed for adversarial settings, where perturbations must evade classifiers while preserving naturalness. We extend these by re-purposing them for black-box transfer attacks.

Preliminary

LDM and DDIM Inversion. LDM performs the iterative denoising process in a compressed latent space for computational efficiency (Rombach et al. 2022). An image $\mathbf{x}_0$ is first mapped to a latent representation $\mathbf{z}_0 = E(\mathbf{x}_0)$ by a pre-trained encoder E. The core of the generative process is a U-Net based noise prediction network $\epsilon_\theta(\mathbf{z}_t, t, C)$, trained to predict the noise added to a latent representation $\mathbf{z}_t$ under a condition C (e.g., text prompt). The key enabler for editing/attacking images is the Denoising Diffusion Implicit Models (DDIM) (Song, Meng, and Ermon 2020), which establishes a deterministic, invertible, non-Markovian denoising path. It finds an initial latent $\mathbf{z}_T$ that can reconstruct $\mathbf{z}_0$ with the following update rule,

$$\mathbf{z}_{t+1} = \sqrt{\bar{\alpha}_{t+1}} \left(\frac{\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{z}_t, t, \varnothing)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t+1}}\epsilon_\theta(\mathbf{z}_t, t, \varnothing) \tag{1}$$

where $\bar{\alpha}_t$ are noise schedule coefficients. The inverted $\mathbf{z}_T$ serves as the starting point for controlled generation.

Single-Point Diffusion-based Attacks. Existing diffusion attacks such as ACA (Chen et al. 2023), DiffAttack (Chen et al. 2024), and Instruct2Attack (Liu et al. 2023) follow a three-stage pipeline:

❶ A clean image $\mathbf{x}_0$ is inverted by DDIM to obtain a latent representation $\mathbf{z}_T$. To ensure high-fidelity reconstruction, techniques like Null-Text Inversion (NTI) (Mokady et al. 2023) are used to optimize the null-text embeddings $\{\varnothing_t\}_{t=1}^T$ to minimize the deviation between the inversion and initial reconstruction trajectories, which yields a tuple $(\mathbf{z}_T, T, C, \{\varnothing_t\}_{t=1}^T)$. C is a conditional prompt embedding.

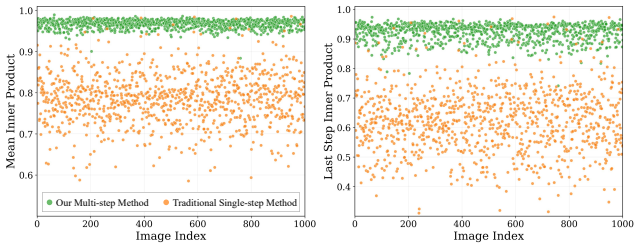
❷ The main step is to introduce a perturbation δ to a single, fixed latent variable, most commonly the initial noise $\mathbf{z}_T$ (Chen et al. 2023),

$$\delta = \arg\max_\delta \mathcal{L}_{\text{CE}}\Big(f\big(D(\hat{\mathbf{z}}_0), y_{\text{gt}}\big)\Big, \quad \hat{\mathbf{z}}_0 = R_{T\to 0}(\mathbf{z}_T + \delta, T, C, \{\varnothing_t\}) \tag{2}$$

where $R_{T\to 0}(\cdot)$ is the full denoising process, $D(\cdot)$ is the decoder, and f is the target classifier.

❸ Finally, during reconstruction, Classifier-Free Guidance (CFG) (Ho and Salimans 2022) is typically used to enforce the semantic guidance from the text prompt C, which is accomplished by interpolating between the conditional and unconditional noise predictions with w controlling the strength of the guidance,

$$\tilde{\epsilon}_\theta(\mathbf{z}_t, t, C) = \epsilon_\theta(\mathbf{z}_t, t, \varnothing_t) + w \cdot \big(\epsilon_\theta(\mathbf{z}_t, t, C) - \epsilon_\theta(\mathbf{z}_t, t, \varnothing_t)\big). \tag{3}$$



(a) Across Steps Inner Product **(b) Last Step Inner Product**

Figure 1: Deviation in noise prediction (measured by the inner product between predicted noise of the clean and perturbed reconstruction path). (a) Averaged inner product across different timesteps; (b) Inner product at the final timestep. Our method (green dots) demonstrates significantly higher stability and consistency in the perturbed noise predictions.

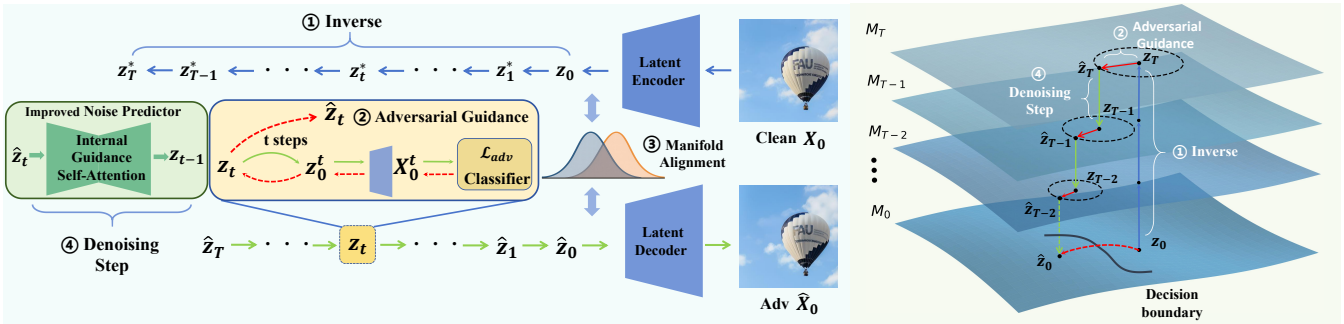


Figure 2: Overview of the proposed HAM framework. ① **Inversion**: A high-fidelity inverse trajectory $\{\mathbf{z}_t^*\}_{t=0}^T$ is first derived from the clean image using a high-order ODE solver; ② **Adversarial Guidance:** At each timestep t, an adversarial gradient is computed using the current latent $\mathbf{z}_t$. This gradient is then projected onto the manifold to create a raw adversarial push δ_t; ③ **Manifold Alignment:** The perturbed latent $\hat{\mathbf{z}}_t = \mathbf{z}_t + \delta_t$ is dynamically aligned with its clean reference $\mathbf{z}_t^*$ by matching their statistical properties (mean and variance), resulting in the final aligned latent $\hat{\mathbf{z}}_t$; ④ **Denoising Step:** A prompt-free, self-attention-guided denoising step is performed on $\hat{\mathbf{z}}_t$ to compute the latent for the next timestep $\mathbf{z}_{t-1}$ and ensure structural consistency, which repeats until the adversarial image is generated.

Methodology

Recall that the existing attacks with static, single-point perturbation has fundamental conflicts with the dynamic, iterative nature of the diffusion process. To validate this empirically, we illustrate in Fig. 1 by measuring the inner product between the noise predictions of a clean reconstruction path and an adversarially perturbed path, $\langle \epsilon_\theta(\mathbf{z}_t^{\text{clean}}, t), \epsilon_\theta(\hat{\mathbf{z}}_t, t)\rangle$. We observe that single-point attacks cause a drastic and accumulating deviation in the predicted noise. Such instability propagates through the denoising steps and becomes magnified at the last step shown by Fig. 1(b), degrading both stealthiness and transferability.

Hierarchical Adversarial Perturbation

Instead of optimizing a single latent $\mathbf{z}_T$, we re-formulate the adversarial objective to iteratively generate the adversarial latent sequence $\{\mathbf{z}_t\}_{t=T}^1$ by finding perturbations $\{\delta_t\}_{t=T}^1$ as a path-wise optimization in each step of the denoising reconstruction,

$$\{\delta_t\}_{t=T}^1 = \underset{\{\delta_t\}_{t=T}^1}{\arg\max} \mathcal{L}_{\text{CE}}\Big(f\big(D(\hat{\mathbf{z}}_0), y_{\text{gt}}\big)\Big), \tag{4}$$

where $\hat{\mathbf{z}}_0$ is the final reconstructed latent of the entire perturbed path. Specifically, at each time step $t \in \{T, T-1, \ldots, 1\}$, we update the latent as,

$$\begin{aligned}\hat{\mathbf{z}}_t &= \mathbf{z}_t + \delta_t \\ &= \mathbf{z}_t + s_t \cdot \nabla_{\mathbf{z}_t} \mathcal{L}_{\text{CE}}\Big(f\big(D\big(R_{t\to 0}(\mathbf{z}_t)\big)\big),\, y_{\text{gt}}\Big),\end{aligned} \tag{5}$$

where $R_{t\to 0}(\cdot)$ is a simulated reconstruction path from the current latent $\mathbf{z}_t$ to $\mathbf{z}_0$ to compute the prospective gradient, and s_t is the perturbation magnitude. At each time step, the perturbation δ_t is adaptively optimized based on the current state to influence the subsequent generative dynamics.However, a caveat with hierarchical perturbations is the error accumulation in high-order solvers due to their dependency on historical steps (Lu et al. 2022). To mitigate this, we introduce a stabilization step. After perturbing $\mathbf{z}_t$ to get $\hat{\mathbf{z}}_t$, we do not use $\hat{\mathbf{z}}_t$ as the direct input for the next iteration. Instead, we use it to predict the noise and then apply the deterministic DDIM update rule to compute the next valid manifold point $\mathbf{z}_{t-1}$,

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{\hat{\mathbf{z}}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta^I(\hat{\mathbf{z}}_t, t)}{\sqrt{\bar{\alpha}_t}}\right)}_{\mathbf{z}_{0|t}^I} + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta^I(\hat{\mathbf{z}}_t, t) \tag{6}$$

where $\epsilon_\theta^I(\hat{\mathbf{z}}_t, t)$ is the noise predicted using our Internal Guidance mechanism, which is detailed in the following subsection. this process ensures that each adversarial "push" is followed by a rectifying step, forcing the trajectory to stay close to the data manifold and thus maintaining path-wise stability throughout the generation process.

Internal Guidance via Unconditional Self-Attention

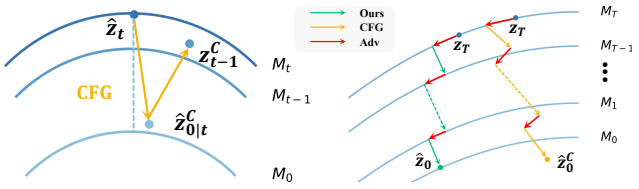


Figure 3: The proposed unconditional guidance corrects manifold deviation in the DDIM denoising process. The transition from $\mathbf{z}_t$ on manifold $\mathcal{M}_t$ to $\mathbf{z}_{t-1}$ relies on predicting the clean image $\mathbf{z}_{0|t}$. Left: CFG introduces a bias due to noisy signal interpolation, causing $\mathbf{z}_{t-1}^c$ to fall off the manifold. Right: The deviation introduced by CFG accumulates over denoising process, while our method provides internal guidance to ensure that each sampling points $\mathbf{z}_t$ remains on the valid manifold.

A primary source of instability in prior work is the reliance on Classifier-Free Guidance (CFG) (Ho and Salimans 2022), which injects text prompt as conditional embeddings into latent feature space via cross-attention and linearly interpolates between the conditional and unconditional predictions (Chen et al. 2024). Nevertheless, ambiguous or misaligned text prompts can introduce conflicting signals that destabilize the generation process and deviate from the manifold (Sarıyıldız et al. 2023). This is illustrated by Fig. 3: because the predicted noise ϵ_θ directly influences the gradient update directions across different manifolds, deviations

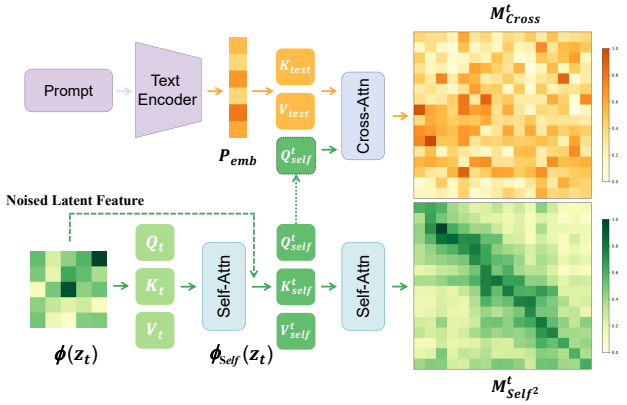


Figure 4: Architectural Comparison for Adversarial Guidance. Top: Standard cross-attention fuses static, external text embeddings P_{emb} with image features, leading to potentially noisy and unstructured guidance maps M^t_{cross}; Bottom: Our cascaded self-attention ($\texttt{SA}^2$) generates a highly structured guidance map ($M^t_{\text{self}^2}$) based solely on time-varying internal features derived from the current noised latent z_t, providing a stable, semantically-consistent guidance signal ϵ^I_θ. Empirical visualizations of cross-attention maps on real images are available in Appendix E.

in noise prediction may cause the updated latent variable $\mathbf{z}_{t-1}$ to deviate from its manifold $\mathcal{M}_{t-1}$. This negatively resonates with the existing methods of performing perturbation searches only on a single manifold corresponding to fixed inversion steps, which not only induces shifts in the overall sampling trajectory but also causes adversarial samples to be trapped in local optima.

To overcome these problems, we propose a prompt-free internal guidance that re-purposes the U-Net's attention blocks at inference time without any retraining. Instead of the standard [Self-Attn $\rightarrow$ Cross-Attn] signal flow, we dynamically re-route the path, composing the self-attention ($\texttt{SA}$) module with itself. This forms a cascaded signal path, [$\text{SA} \rightarrow \text{SA}^2$], as illustrated in Fig. 4. We formulate this process as follows, where $\phi(\mathbf{z}_t)$ is the feature map entering the attention block:

$$\phi_{\text{self}}(\mathbf{z}_t) = \text{Self-Attn}\big((W_Q, W_K, W_V) \cdot \phi(\mathbf{z}_t)\big) + \phi(\mathbf{z}_t) \quad (7)$$

$$Q^t_{\text{self}}, K^t_{\text{self}}, V^t_{\text{self}} = (W_Q, W_K, W_V) \cdot \phi_{\text{self}}(\mathbf{z}_t) \quad (8)$$

$$\mathbf{M}^t_{\text{self}^2} = \text{Softmax}\left(\frac{Q^t_{\text{self}}(K^t_{\text{self}})^\top}{\sqrt{d}}\right) \quad (9)$$

$$\phi_{\text{self}^2}(\mathbf{z}_t) = (\mathbf{M}^t_{\text{self}^2})V^t_{\text{self}} + \phi_{\text{self}}(\mathbf{z}_t). \quad (10)$$

While self-attention has been leveraged for tasks like text-guided image editing, often by re-assembling the attention layer (Tumanyan et al. 2023; Cao et al. 2023), our approach introduces a distinct application by leveraging self-attention in the adversarial domain:

$$\epsilon^I_\theta(\hat{\mathbf{z}}_t, t) \triangleq \epsilon_\theta(\hat{\mathbf{z}}_t, t; \text{flow: SA} \rightarrow \text{SA}^2) \quad (11)$$

This cascaded application functions as a hierarchical repairing mechanism that the first pass establishes a global structural context, and in turn, the second pass leverages this context to refine local details and correct inconsistencies introduced by adversarial perturbations. The resulting positive feedback loop ensures internal consistency that enables natural integration of adversarial signals without conflicting external prompts as other methods (Chen et al. 2023, 2024). In practice, we leverage the insight that self-attention dominates the later stages of diffusion to refine local details and textures (Zhang et al. 2024), and strategically initiate the hierarchical attack from an intermediate step (e.g., starting from $T \leq N_{inv}/2$). This strategy not only focuses the adversarial budget on more fine-grained semantic manipulations but also yields significant computational benefits by circumventing the now-redundant cross-attention computations, as shown in Table 3.

Dynamic Manifold Alignment

While the hierarchical framework and stable guidance mitigates trajectory deviation, the iterative nature of perturbations introduces a more fundamental challenge of distributional drift. Each adversarial step pushes the latent variable $\hat{\mathbf{z}}_t$ slightly away from its corresponding data manifold $\mathcal{M}_t$. This drift accumulates and makes $\hat{\mathbf{z}}_t$ an Out-of-Distribution (OOD) sample for the denoising network $\boldsymbol{\epsilon_\theta}$. This violation of the model's core training assumption—that inputs are statistically consistent with the forward process $q(\mathbf{z}_t \mid \mathbf{z}_0)$—leads to inaccurate noise predictions and unnatural visual artifacts (Lin et al. 2024).

Inspired by research in style transfer (Huang and Belongie 2017), we propose Dynamic Manifold Alignment, a distributional correction mechanism integrated within each attack step. Instead of imposing a new style, we introduce a *minimal, corrective force* that nullifies statistical shifts induced by the adversarial gradient. This force is a nearly cost-free moment matching operation—unlike iterative or loss-based alignment—that analytically projects the OOD latent $\hat{\mathbf{z}}_t$ back to its true manifold's statistical center. This is achieved by re-anchoring the statistics of the adversarial latent to a reliable, dynamic baseline of the corresponding clean latent $\mathbf{z}^*_t$. This latent serves as an ideal, empirically on-manifold reference that the denoising network is conditioned to handle correctly. This alignment is performed after each adversarial gradient step and immediately before the DDIM update,

$$\hat{\mathbf{z}}_t = \mu(\mathbf{z}^*_t) + \sigma(\mathbf{z}^*_t) \odot \frac{\hat{\mathbf{z}}_t - \mu(\hat{\mathbf{z}}_t)}{\sigma(\hat{\mathbf{z}}_t)}, \quad t \in \{T, T-1, \ldots, 1\} \quad (12)$$

where the structure of the adversarial latent $\frac{\hat{\mathbf{z}}_t - \mu(\hat{\mathbf{z}}_t)}{\sigma(\hat{\mathbf{z}}_t)}$ is preserved while its channel-wise statistics are replaced by those of the clean reference. It ensures the denoising network receives a statistically aligned input, that decouples the disruptive goal of adversarial perturbation from the constructive requirement of generative stability, and preserves the fidelity of the denoising process throughout multi-step perturbation.

Manifold-Constrained Optimization

Finally, we detail the core of the perturbation step δ_t. Rather than a simple gradient ascent, we design a manifold-aware update that incorporates momentum, tangent space projection, and adaptive scaling to ensure stable and effective optimization. The full update is:

$$\hat{\mathbf{z}}_t = \mathbf{z}_t + s_t \cdot \text{sign}\big(\Pi_\perp(g_t)\big) \quad (13)$$

in which g_t is the momentum-integrated gradient, s_t is the adaptive perturbation magnitude and $\Pi_\perp(\cdot)$ is a projection onto the manifold's tangent space.

Manifold-projected Momentum Gradient. To restrict $\hat{\mathbf{z}}_t$ on valid manifold regions, we enhance the standard gradient in two ways: a) incorporating momentum; b) projection on data manifold. We use momentum to stabilize the optimization trajectory by accumulating historical gradients,

$$g_t \leftarrow m \cdot g_{t+1} + \nabla_{\mathbf{z}_t} \mathcal{L}_{\text{CE}}\Big(f\big(D\big(R_{t\to 0}(\mathbf{z}_t)\big)\big),\, y_{\text{gt}}\Big) \tag{14}$$

where m is the momentum decay factor. Further, we constrain this gradient to the manifold's tangent space. Based on the insight that the predicted score $\mathbf{s}_\theta(\mathbf{z}_t, t)$ (or its proxy $\epsilon_\theta(\mathbf{z}_t, t)$) is orthogonal to the data manifold $\mathcal{M}_t$ (Song et al. 2020), we treat it as the manifold's normal vector, $\nabla_{\text{normal}} = \epsilon_\theta(\mathbf{z}_t, t)$. To prevent the adversarial update from corrupting the denoising direction, we project g_t to be orthogonal to this normal vector:

$$\Pi_\perp(g_t) \leftarrow g_t - \left(\frac{g_t \cdot \nabla_{\text{normal}}}{\|\nabla_{\text{normal}}\|_2^2}\right) \nabla_{\text{normal}} \tag{15}$$

Adaptive Perturbation Magnitude. The noise level varies significantly with timestep t, determined by the noise schedule $\bar{\alpha}_t$. A fixed perturbation magnitude would be too aggressive in low-noise steps and vice versa. Hence, we design an adaptive strategy to scale perturbation strength relative to the noise intensity,

$$s_t = s_T \cdot \sqrt{\frac{1-\bar{\alpha}_t}{1-\bar{\alpha}_T}} \tag{16}$$

where s_T is the initial adversarial perturbation magnitude, and $\bar{\alpha}_t$ and $\bar{\alpha}_T$ represent the noise schedule coefficients at t and the starting T, respectively. This adaptive mechanism ensures that our adversarial guidance is consistently scaled with the manifold's geometric characteristic at each step.

Putting it All Together

The entire framework is summarized in Algorithm 1. The attack consists of two phases. First, an inversion process obtains the clean reference path $\{\mathbf{z}_t^*\}$. Second, the hierarchical reconstruction generates the adversary. At each step of this phase, it (1) computes a manifold-constrained gradient, (2) applies the perturbation, (3) re-aligns statistics to the clean path, and (4) performs a denoising step using the proposed prompt-free internal guidance.

Theoretical Analysis

In this section, we provide a theoretical understanding to the hierarchical adversarial attacks based on (Chung et al. 2022), and detail the assumptions, preliminary propositions and proofs in Appendix B.

Theorem 1 ***(Tangent Alignment of Adversarial Gradients)*** *Let the multi-step DDIM reconstruction from timestep $t \to 0$ be denoted by $R_{t\to 0}(\mathbf{z}_t)$, and let the adversarial loss be $L = \mathcal{L}_{CE}(f(D(R_{t\to 0}(z_t))), y_{gt})$. Under Assumption 1-2 and Propositions B.1, B.2 (Appendix B), the backpropagated gradient satisfies,*

$$\left\|\Pi_{\mathcal{N}_{\mathbf{z}_t}\mathcal{M}_t}\left(\nabla_{\mathbf{z}_t} L\right)\right\|_2 \ll \left\|\Pi_{\mathcal{T}_{\mathbf{z}_t}\mathcal{M}_t}\left(\nabla_{\mathbf{z}_t} L\right)\right\|_2 \tag{17}$$

Algorithm 1: Hierarchical Attack with Manifold-Awareness

Input: Input image X_0 with true label y_{gt}, classifier $f(\cdot)$, latent encoder $E(\cdot)$ and decoder $D(\cdot)$, reconstruction starting point T, adaptive perturbation magnitude s_t.$\mathbf{z}_0^* = E(X_0)$

Output: Adversarial image $\hat{X}_0$

for $t = 0$ **to** $T-1$ **do**
  $\mathbf{z}_t^* \leftarrow \mathbf{z}_{t+1}^*$ with two-order ODE solvers // Phase 1: Latent Image Inversion
end for
$\mathbf{z}_T \leftarrow \mathbf{z}_T^*, g_T \leftarrow 0$ // Phase 2: Hierarchical Adversarial Reconstruction
for $t = T \to 1$ **do**
  for $k = t \to 1$ **do**
    $\mathbf{z}_{k-1}^t \leftarrow R_{k\to k-1}(\mathbf{z}_k^t, \epsilon_\theta)$ // Eq. 6
  end for
  $g_t \leftarrow m \cdot g_{t-1} + \nabla_{\mathbf{z}_t}\mathcal{L}_{\text{CE}}\Big(f\big(D\big(\mathbf{z}_0^t\big)\big),\, y_{\text{gt}}\Big)$ // Eq. 14
  $\hat{\mathbf{z}}_t \leftarrow \mathbf{z}_t + s_t \cdot \text{sign}\left(\Pi_\perp(g_t)\right)$ // Eqs. 15,16
  $\hat{\mathbf{z}}_t = \mu(\mathbf{z}_t^*) + \sigma(\mathbf{z}_t^*) \odot \frac{\hat{\mathbf{z}}_t - \mu(\hat{\mathbf{z}}_t)}{\sigma(\hat{\mathbf{z}}_t)}$ // Eq. 12
  $\epsilon_\theta^I(\hat{\mathbf{z}}_t, t) \triangleq \epsilon_\theta(\hat{\mathbf{z}}_t, t;$ flow: SA $\to$ SA$^2)$ // Eqs. 7,8,9,10,11
  $\mathbf{z}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\hat{\mathbf{z}}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta^I}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta^I$ // Eq. 6
end for
return $\hat{X}_0 \leftarrow D(\hat{\mathbf{z}}_0)$

where $Proj_{\mathcal{T}_{\mathbf{z}_t}\mathcal{M}_t}$ is the projection onto the tangent space $\mathcal{T}_{\mathbf{z}_t}\mathcal{M}_t$ of the noisy data manifold $\mathcal{M}_t$.

Theorem 1 states that the backpropagated adversarial gradient is approximately confined to the local tangent space of the noisy data manifold, i.e., for an attack to be effective and stable, its perturbations must align with the intrinsic geometry of the generative process, as any signal in the normal direction is actively suppressed by the model's own denoising mechanism. This leads to the next corollary that motivates the design of dynamic manifold alignment.

Corollary 1 ***(Manifold Alignment)*** *The tangent space $\mathcal{T}_{\mathbf{z}_t}\mathcal{M}_t$ evolves across time steps according to*

$$\mathcal{T}_{\mathbf{z}_{t-1}}\mathcal{M}_{t-1} \neq J_{g_t}^T(\mathcal{T}_{\mathbf{z}_t}\mathcal{M}_t), \tag{18}$$

where J_{g_t} is the Jacobian of the denoising step.

Thus, projecting the adversarial gradient onto the tangent space suffers from manifold drift caused by cumulative errors so we must continuously update our tangent space estimates through dynamic manifold alignment to track the evolving manifold geometry. Appendix E details a directional analysis between the adversarial gradient g_t and the predicted score ϵ_θ before the final projection (15).

Experiments

Experimental Setup

Dataset and Models. Similar to (Zhao, Liu, and Larson 2020; Yuan et al. 2022; Chen et al. 2023), we evaluate on the ImageNet-Compatible dataset (Russakovsky et al. 2015). To evaluate adversarial transferability across architectures, we select representative models from Convolutional Neural Networks, Vision Transformers, and Multilayer Perceptrons. Specifically, these include: ResNet-50 (He et al.

Surrogate Model	Attacks	CNNs				Vision Transformers				MLPs		Avg ↑ ASR(%)	FID ↓	LPIPS ↓
		Res-50	VGG-19	Mob-v2	Inc-v3	ViT-B	Swin-B	Deit-B	Deit-S	Mix-B	Mix-L			
	Clean	7.3	11.3	13.1	19.5	6.3	4.1	5.5	6.0	17.5	23.5	11.87	57.8	0.000
Res-50	DIM	95.3	84.1	75.2	70.7	19.9	33.8	20.8	23.6	38.0	42.8	45.43	77.4	0.411
	TIM	89.5	71.1	69.7	46.0	25.6	16.7	20.2	25.3	39.4	43.8	39.76	82.2	0.299
	cAdv	97.4	36.2	43.9	24.6	27.7	20.5	26.1	30.7	39.9	48.1	33.08	64.3	0.159
	NCF	89.8	72.0	72.3	33.4	37.8	27.0	28.8	36.3	47.0	54.8	45.49	72.4	0.416
	ACA	81.4	66.5	69.8	63.4	60.0	59.9	59.9	58.9	67.1	69.6	63.92	71.2	0.725
	DiffAttack	96.4	73.3	76.6	65.7	49.7	56.4	49.6	55.8	57.6	59.2	60.43	62.9	0.237
	HAM*	95.2	76.8	81.1	69.5	59.1	60.1	58.9	62.8	67.2	64.2	**66.63**	**61.9**	**0.197**
Mob-v2	DIM	63.6	83.1	96.6	62.0	16.9	27.0	16.6	20.4	35.6	41.2	40.71	75.5	0.412
	TIM	58.6	76.1	93.5	44.5	23.1	17.0	19.2	26.1	41.2	45.0	38.98	84.0	0.306
	cAdv	38.7	40.9	97.0	25.1	28.0	19.6	24.3	30.1	40.4	49.2	32.92	65.5	0.174
	NCF	64.8	72.5	92.5	34.8	34.2	25.7	25.6	33.6	46.2	51.6	43.22	71.4	0.417
	ACA	64.2	64.1	84.6	60.5	57.3	56.0	58.2	57.8	67.7	69.0	61.64	69.2	0.724
	DiffAttack	76.3	77.2	98.2	61.5	47.3	53.6	45.4	54.3	58.6	60.1	59.37	64.1	0.240
	HAM*	81.9	79.7	97.2	69.4	60.4	59.1	56.6	64.9	68.2	68.4	**67.62**	**63.0**	**0.206**
Inc-v3	DIM	60.3	69.6	66.6	96.2	22.4	26.5	22.4	25.6	39.1	46.2	42.08	79.1	0.392
	TIM	50.2	58.4	56.9	94.2	25.8	17.5	21.3	26.0	38.9	44.0	37.67	72.4	0.289
	cAdv	19.4	24.7	26.3	25.4	15.9	14.0	15.1	17.3	32.2	40.4	22.81	61.9	0.136
	NCF	48.3	58.5	56.5	41.2	26.1	17.3	17.8	25.2	39.5	46.0	37.24	67.7	0.375
	ACA	63.2	63.5	64.9	80.0	57.4	57.1	60.3	58.4	66.7	69.6	62.34	70.3	0.713
	DiffAttack	39.4	41.2	42.9	74.9	23.8	25.4	25.4	25.0	42.0	45.1	34.46	62.2	0.219
	HAM*	64.8	64.7	64.4	98.0	47.4	47.3	45.9	48.9	59.9	61.8	**56.12**	**64.7**	**0.204**
Swin-B	DIM	45.6	58.2	54.0	44.8	29.1	84.1	33.8	33.3	44.3	47.2	43.37	66.1	0.378
	TIM	39.3	52.9	51.5	31.6	34.8	56.5	35.5	35.3	42.2	46.1	41.02	67.1	0.304
	cAdv	30.1	29.6	35.5	21.9	35.1	96.8	40.7	41.6	42.2	50.4	36.34	63.4	0.146
	NCF	50.4	56.2	56.0	26.6	37.6	63.7	35.4	38.8	46.9	51.0	44.32	66.6	0.374
	ACA	62.5	63.2	64.8	57.5	61.1	73.8	63.2	61.3	65.7	69.7	63.22	68.6	0.731
	DiffAttack	56.6	55.9	58.6	51.6	60.4	90.4	65.0	62.8	63.9	62.4	59.69	65.8	0.239
	HAM*	59.9	59.5	63.5	54.9	63.8	94.4	68.1	66.7	66.4	67.7	**63.39**	**64.6**	**0.202**

Table 1: Comparison of attack performance across different model architectures. For statistical robustness, all reported metrics are averaged over 5 independent runs (see Appendix D for details). AVG denotes the average accuracy on all the models except the one that same as the surrogate. Higher ASR (%) indicates better attack performance, while lower FID indicates better perceptual quality.

Method	HGD	R&P	NIP-r3	DiffPure	Adv-Inc-v3	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Avg ↑
DIM	26.6	16.5	15.6	26.8	25.3	19.9	19.2	9.7	19.95
TIM	35.1	34.9	32.0	54.3	33.0	36.2	39.4	29.6	36.81
cAdv	19.8	24.6	20.5	56.2	24.2	25.3	27.8	20.4	27.35
NCF	34.3	38.7	32.1	84.1	34.7	36.2	38.3	30.8	41.15
ACA	57.4	58.5	59.2	69.4	61.7	64.2	65.6	60.7	62.09
DiffAttack	60.5	51.5	55.9	78.0	62.1	60.8	62.6	54.4	60.73
HAM*	**69.6**	**59.9**	**61.4**	**83.9**	**68.0**	**67.0**	**66.6**	**60.5**	**67.11**

Table 2: Comparison of attack performance across different adversarial defense approaches Metrics are averaged over 5 independent runs for robustness.

Figure 5: Visual artifacts of different attack methods.

Attack	cAdv	NCF	ACA	DiffAttack	HAM*
Time (sec)	22.4	18.7	>200	29.9	21.6

Table 3: Comparison of computational cost (seconds/image). HAM achieves high efficiency by mitigating complex initial optimization and single-point iterative loops.

2016), VGG-19 (Simonyan and Zisserman 2014), Inception-v3 (Szegedy et al. 2016), and MobileNet-v2 (Sandler et al. 2018); ViT-B/16 (ViT-B) (Dosovitskiy et al. 2020), Swin-B (Liu et al. 2021), DeiT-B, and DeiT-S (Touvron et al. 2021); and Mixer-B/16 (Mix-B) and Mixer-L/16 (Mix-L) (Tolstikhin et al. 2021).

Implementation Details. All experiments are implemented on a single RTX3090 GPU. The total number of inverse steps N_{inv} is set to 30, with the reconstruction starting point $T = 12$, i.e., the reconstruction begins after T inversion steps from the original image. During reconstruction, we set the momentum coefficient $m = 0.9$ and the initial perturbation magnitude $s_T = 0.035$.

Evaluation Metrics. We adopt Attack Success Rate (ASR) to measure the attack effectiveness, and quantify the perceptual quality of generated adversarial samples at the human visual level using the Fréchet Inception Distance (FID) (Heusel et al. 2017) and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018).

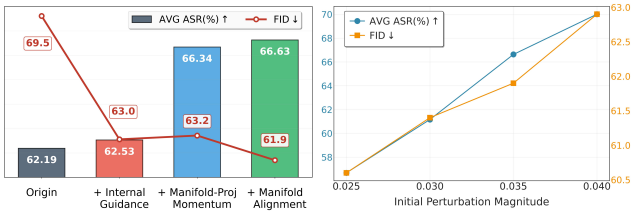


Figure 6: Ablation studies. Left: effectiveness of individual components; Right: Impact of initial perturbation magnitude. A detailed sensitivity analysis for the key hyperparameters, including the reconstruction starting point and the momentum coefficient, are available in Appendix D.

Performance Comparison

Attacks on Normally Trained Models. We compare with a wide range of SOTA attacks, including traditional transferability-based DIM (Xie et al. 2019) and TIM (Dong et al. 2019) with pixel-level L_p-norm constraints, as well as unrestricted attack methods cAdv (Bhattad et al. 2019), NCF (Yuan et al. 2022), ACA (Chen et al. 2023) and DiffAttack (Chen et al. 2024). For all baseline methods, we adopted their official implementations and the optimal parameter settings reported in their original papers. We adopt Res-50, Mob-v2, Inc-v3, and Swin-B as the surrogate models to generate adversarial examples. Table 1 presents the main results of attack transferability and imperceptibility, where the identical surrogate and target models represent white-box attacks (highlighted in gray), otherwise the attacks are black-box. Our proposed method exhibits superior transferability and imperceptibility across diverse model architectures. For attack transferability, compared to DiffAttack, HAM achieves substantial improvements of ASR with 10.2%, 13.9%, 62.8%, and 6.2% on the four surrogate models, respectively. Compared to other baseline methods, it achieves an average improvements over 30%. Regarding visual fidelity, compared to existing state-of-the-art methods, the proposed method achieves improvements of 1.0-1.2 on the FID metric and 6.8%-16.7% on the LPIPS metric. This highlights its significant advantages in balancing visual quality and attack performance.

Fig. 5 visualizes the adversarial examples generated by different attack methods. The adversarial examples produced by DIM and TIM exhibit conspicuous high-frequency noise patterns. Unrestricted attack methods tend to introduce unnatural color shifts (cAdv, NCF), semantic distortions (DiffAttack) and even fundamental image alterations(ACA). In contrast, HAM demonstrates superior performance in preserving both visual naturalness and semantic consistency. More visual results are available in the Appendix E.

Attacks Against Defense Methods. To evaluate the effectiveness of different attack methods against existing defense mechanisms, we select two representative types of defenses: input pre-processing defenses (HGD (Liao et al. 2018), R&P (Xie et al. 2017), NIPS-r3 (Thomas and Elibol 2017) and DiffPure (Nie et al. 2022)) and adversarially trained models (Adv-Inc-v3 (Kurakin, Goodfellow, and Bengio 2018), Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ (Tramèr et al. 2017)). All use ResNet-50 as the surrogate model. Table 2 presents the black-box transferability results under different defense mechanisms. HAM achieves significantly higher averageachieves significantly higher average ASR compared to existing methods across all defense mechanisms, with an improvement of 10.5% compared to SOTA. This highlights HAM's superior performance against defended models.

Ablation Studies

We also conduct ablation studies to assess component-wise contribution to the adversarial objective in Fig. 6. We use ResNet-50 and multi-step perturbation based on CFG as the "Origin". The results indicate that the incorporation of self-attention as an internal guidance plays a pivotal role in enhancing the imperceptibility (sharp drop of FID from 69.7 to 62.9), which enhances the stability of the image generation process during the reconstruction by rectifying prediction directional bias, though it does not improve transferability alone; then, manifold-projected momentum mechanism significantly improves the attack transferability by stabilizing the optimization trajectory with momentum and manifold tangent space projection on the adversarial guidance gradients; finally, the dynamic step-wise alignment mechanism tightly aligns the reconstruction trajectory with the natural image distribution, giving the overall performance an ultimate boost in both ASR and visual quality.

Conclusion

In this work, we propose a novel diffusion-based attack that overcomes the limitations of single-point perturbations. By leveraging unconditional self-attention for stable guidance and dynamic alignment for manifold regularization, HAM achieves superior performance in adversarial transferability, efficiency, and image quality.

References

Bhattad, A.; Chong, M. J.; Liang, K.; Li, B.; and Forsyth, D. A. 2019. Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*.

Cao, M.; Wang, X.; Qi, Z.; Shan, Y.; Qie, X.; and Zheng, Y. 2023. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, 22560–22570.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. Ieee.

Chen, J.; Chen, H.; Chen, K.; Zhang, Y.; Zou, Z.; and Shi, Z. 2024. Diffusion models for imperceptible and transferable adversarial attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen, Z.; Li, B.; Wu, S.; Jiang, K.; Ding, S.; and Zhang, W. 2023. Content-based unrestricted adversarial attack. *Advances in Neural Information Processing Systems*, 36: 51719–51733.

Chung, H.; Sim, B.; Ryu, D.; and Ye, J. C. 2022. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35: 25683–25696.

Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4312–4321.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Duan, R.; Chen, Y.; Niu, D.; Yang, Y.; Qin, A. K.; and He, Y. 2021. Advdrop: Adversarial attack to dnns by dropping information. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7506–7515.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, 1501–1510.

Inkawhich, N.; Wen, W.; Li, H. H.; and Chen, Y. 2019. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7066–7074.

Jia, S.; Yin, B.; Yao, T.; Ding, S.; Shen, C.; Yang, X.; and Ma, C. 2022. Adv-attribute: Inconspicuous and transferable adversarial attack on face recognition. *Advances in Neural Information Processing Systems*, 35: 34136–34147.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, 99–112. Chapman and Hall/CRC.

Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; and Zhu, J. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1778–1787.

Lin, S.; Liu, B.; Li, J.; and Yang, X. 2024. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 5404–5411.

Liu, J.; Wei, C.; Guo, Y.; Yu, H.; Yuille, A.; Feizi, S.; Lau, C. P.; and Chellappa, R. 2023. Instruct2attack: Language-guided semantic adversarial attacks. *arXiv preprint arXiv:2311.15551*.

Liu, L.; Ren, Y.; Lin, Z.; and Zhao, Z. 2022. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Long, Y.; Zhang, Q.; Zeng, B.; Gao, L.; Liu, X.; Zhang, J.; and Song, J. 2022. Frequency domain model augmentation for adversarial attack. In *European conference on computer vision*, 549–566. Springer.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35: 5775–5787.

Lu, Y.; Jia, Y.; Wang, J.; Li, B.; Chai, W.; Carin, L.; and Velipasalar, S. 2020. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 940–949.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mokady, R.; Hertz, A.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2023. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6038–6047.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*.

Qiu, H.; Xiao, C.; Yang, L.; Yan, X.; Lee, H.; and Li, B. 2020. Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In *European conference on computer vision*, 19–37. Springer.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Sarıyıldız, M. B.; Alahari, K.; Larlus, D.; and Kalantidis, Y. 2023. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8011–8021.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Shu, R.; Kushman, N.; and Ermon, S. 2018. Constructing unrestricted adversarial examples with generative models. *Advances in neural information processing systems*, 31.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Thomas, A.; and Elibol, O. 2017. Defense against adversarial attacks - 3rd place.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, 10347–10357. PMLR.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Tumanyan, N.; Geyer, M.; Bagon, S.; and Dekel, T. 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1921–1930.

Xiao, C.; Zhu, J.-Y.; Li, B.; He, W.; Liu, M.; and Song, D. 2018. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*.

Xiao, Y.; and Wang, C. 2021. You see what I want you to see: Exploring targeted black-box transferability attack for hash-based image retrieval systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1934–1943.

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2017. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2730–2739.

Yang, X.; Zhou, D.; Feng, J.; and Wang, X. 2023. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 22552–22562.

Yuan, S.; Zhang, Q.; Gao, L.; Cheng, Y.; and Song, J. 2022. Natural color fool: Towards boosting black-box unrestricted attacks. *Advances in Neural Information Processing Systems*, 35: 7546–7560.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhang, W.; Liu, H.; Xie, J.; Faccio, F.; Shou, M. Z.; and Schmidhuber, J. 2024. Cross-attention makes inference cumbersome in text-to-image diffusion models. *arXiv e-prints*, arXiv–2404.

Zhao, Z.; Liu, Z.; and Larson, M. 2020. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1039–1048.

Reproducibility Checklist

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) yes

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) NA

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) NA

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) partial

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) partial

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes