VISDELTA: MODEL INVERSION FROM RELEASED LORA ADAPTERS FOR VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-Rank Adaptation (LoRA) is a widely used parameter-efficient fine-tuning (PEFT) method for vision-language models (VLMs) that updates only a small fraction of their parameters. When trained on private images, LoRA adapters retain visual information from them. Adapters are commonly shared on public model platforms, creating a serious risk of model inversion. In this paper, we propose **VisDelta**, a model inversion attack against released VLM LoRA adapters. It compares the base and LoRA-adapted VLMs on the same image and query and uses their response difference to guide image reconstruction. Firstly, VisDelta samples latent codes at random and generates an initial image from each using a public image generator. It ranks these images by an adapter-aware score, retains the codes that produced the top-ranked images, and refines each code through adapter-aware differential guidance to reconstruct private images associated with the target. The guidance combines target agreement, separation from hard-negative conditions, and the adapter-induced gain in target separation, while keeping each code close to its initial value. Refinement uses random image transformations; final selection balances average and worst-view losses over fresh transformations to choose a reconstruction with stable target responses. Experiments across four VLMs and three benchmark datasets show that VisDelta recovers target-related visual content from LoRA adapters trained on private images. On CelebA with Qwen2.5-VL-7B, it achieves 26.38% Acc@1 and 52.47% Acc@5, exceeding the strongest baseline by 3.81 and 2.29 percentage points, respectively. These results show that released VLM LoRA adapters can expose private visual information.

1 INTRODUCTION

Vision-language models (VLMs) jointly process images and text (Radford et al., 2021), enabling a single model to interpret visual content and answer questions in natural language (Alayrac et al., 2022; Li et al., 2022; 2023b; Liu et al., 2023). They support applications in autonomous driving, medical image analysis, and visual content moderation (Sima et al., 2024; Li et al., 2023a; Helff et al., 2025). Adapting a large VLM through full fine-tuning, however, requires substantial computation and storage. Parameter-efficient fine-tuning (PEFT) reduces this cost by freezing most pretrained parameters and learning only a small update for each task (Houlsby et al., 2019; Li & Liang, 2021; Lester et al., 2021). Low-Rank Adaptation (LoRA) parameterizes this update with low-rank matrices and has become a widely used PEFT method (Hu et al., 2022; Dettmers et al., 2023). The resulting adapters are compact, easy to distribute, and compatible with public base models. LoRA adapters are now widely shared on public model platforms, making their learned updates accessible to others. When trained on private images, these updates can retain visual information from the training data, creating a model inversion risk when the adapter weights are released publicly.

Recent studies examine leakage from LoRA-adapted language models (Ran et al., 2025), inversion of training-time adapter gradients (Sami et al., 2025), leakage from LoRA-fine-tuned diffusion models (Yao, 2024), and extraction of adapted functionality (Wang et al., 2025). They leave open whether a released VLM LoRA adapter retains enough visual information for image reconstruction. Figure 1 illustrates the threat: the attacker has the public base VLM and released adapter but no private training images or training-time information. By comparing the base and LoRA-adapted VLMs on the same image and query, the attacker seeks to reconstruct visual content associated with a target condition, or another fine-grained sensitive attribute.

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.

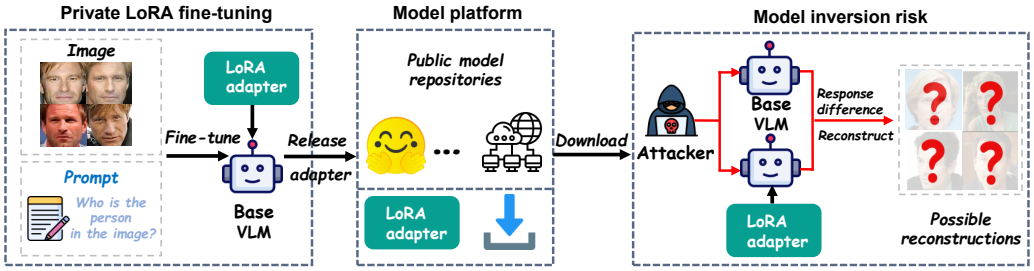

Figure 1: **Model inversion risk from released VLM LoRA adapters.** A victim fine-tunes a LoRA adapter for a public base VLM using private image–prompt pairs and releases it through a public model repository. After downloading the adapter, an attacker compares the VLM responses before and after loading it and uses their difference as a reconstruction signal. The face recognition example illustrates the resulting risk to visual information contained in the private fine-tuning data.

Existing model inversion methods do not isolate the information contributed by a released LoRA adapter. Classical attacks optimize the latent code of a public image generator until a discriminative model predicts the target class (Fredrikson et al., 2015; Zhang et al., 2020; Struppek et al., 2022). Recent work extends model inversion to VLMs by optimizing an image with answer losses at the token or sequence level (Nguyen et al., 2026). These methods treat the victim VLM as a single model. When applied to an adapted VLM, their objectives cannot distinguish knowledge already present in the base model from changes introduced by LoRA. This attribution gap is central to inversion from released adapters, because a strong target response may arise from the base VLM rather than information learned from the private fine-tuning data.

Challenges. Model inversion from a released VLM LoRA adapter presents two coupled challenges. First, the adapted model combines knowledge already present in the base VLM with changes introduced by LoRA. Optimizing the adapted model alone may therefore recover generic target cues rather than visual information encoded during private fine-tuning. Second, the attack must search the nonconvex latent space of a public generator. Random initializations may provide little target evidence, while optimizing a single view can exploit unstable cues that vanish under benign transformations. An effective attack must isolate LoRA-induced changes, identify informative initializations, and recover visual content that remains stable across views.

Our approach. The base VLM provides a reference for isolating the effect of the released adapter. For the same generated image and query, VisDelta compares target margins before and after loading LoRA and uses their difference as an adapter-specific reconstruction signal. It first samples latent codes from a public image generator and ranks the resulting images. The top-ranked codes are refined through **adapter-aware differential guidance**, which promotes target agreement, suppresses alternative conditions, and favors the increase in target separation induced by LoRA. A latent-space constraint limits drift from each initialization. Refinement uses random image transformations, and final selection combines the mean and worst-view guidance losses over fresh views.

In summary, we make the following contributions:

- We demonstrate a model inversion risk in released VLM LoRA adapters by recovering target-related visual information learned from private fine-tuning images.
- We propose **VisDelta**, which ranks images generated from a public image generator and refines their latent codes through adapter-aware differential guidance based on changes between the base and LoRA-adapted VLMs. Random transformations during refinement and selection across fresh views improve reconstruction consistency.
- We evaluate VisDelta across four VLMs and three benchmark datasets. On CelebA with Qwen2.5-VL-7B, it achieves 26.38% Acc@1 and 52.47% Acc@5, exceeding the strongest baseline by 3.81 and 2.29 percentage points, respectively.

2 Related Work

Parameter-efficient adaptation of VLMs. Parameter-efficient fine-tuning (PEFT) adapts pre-trained models while updating only a small fraction of their parameters. Prompt-learning methods

optimize textual, visual, or multimodal context tokens while keeping the backbone fixed (Lester et al., 2021; Zhou et al., 2022b;a; Khattak et al., 2023). Adapter methods insert lightweight trainable modules into visual, language, or cross-modal pathways (Houlsby et al., 2019; Sung et al., 2022; Gao et al., 2024; Luo et al., 2023). Low-rank methods instead represent weight updates using compact trainable matrices, as in LoRA and QLoRA (Hu et al., 2022; Dettmers et al., 2023). This literature primarily evaluates adaptation accuracy and parameter efficiency. We examine whether a VLM LoRA adapter released as a separate artifact retains private visual information from its fine-tuning data.

Privacy risks of LoRA adapters. Prior work studies membership leakage in VLMs (Hu et al., 2025) and training-data extraction or replication in language and diffusion models (Carlini et al., 2021; 2023; Somepalli et al., 2023). Gradient inversion recovers training data from exposed gradients (Zhu et al., 2019; Yin et al., 2021), including adapter gradients during PEFT (Sami et al., 2025). For released adapters, LoRA-Leak investigates language-model membership inference (Ran et al., 2025); diffusion LoRA can reveal private visual content (Yao, 2024), while StolenLoRA extracts task capabilities (Wang et al., 2025). VisDelta instead reconstructs target-related visual content from a final VLM LoRA adapter using the public base model.

Model inversion and visual reconstruction. Model inversion reconstructs sensitive attributes or representative inputs from model behavior (Fredrikson et al., 2015). Modern attacks use a public image generator as a visual prior and optimize its latent variables toward a target class (Zhang et al., 2020; Wang et al., 2021; An et al., 2022; Struppek et al., 2022; Yuan et al., 2023). Recent work extends model inversion to autoregressive VLMs through objectives defined over answer tokens or complete answer sequences (Nguyen et al., 2026). These methods treat the victim as a single trained model and therefore cannot distinguish knowledge already present in the base model from changes introduced by LoRA. VisDelta instead compares the responses of the same VLM before and after loading the adapter and uses their difference to guide visual reconstruction.

3 Problem Setup and Threat Model

Let $M_0 = f_{\theta_0}$ denote a public base vision-language model (VLM). Let $\mathcal{C}$ be the set of target conditions, and let $y(c) = (y_1^c, \ldots, y_{T_c}^c)$ denote the answer sequence associated with condition $c \in \mathcal{C}$. The victim fine-tunes M_0 on a private dataset

$$\mathcal{D}_{\text{priv}} = \{(x_i, q_i, c_i)\}_{i=1}^n , \tag{1}$$

where x_i is an image, q_i is a task query, and $c_i \in \mathcal{C}$ is the corresponding target condition.

Let $\Delta\theta$ denote the effective parameter update induced by the learned LoRA factors. The victim keeps θ_0 fixed and learns the adapter by minimizing the length-normalized answer loss

$$\mathcal{L}_{\text{ft}}(\Delta\theta) = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{T_{c_i}}\sum_{r=1}^{T_{c_i}} \log p_{\theta_0+\Delta\theta}\left(y_r^{c_i} \mid y_{<r}^{c_i}, x_i, q_i\right). \tag{2}$$

For an adapted linear layer with base weight $W_0 \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$, LoRA produces the adapted weight

$$W_1 = W_0 + \Delta W, \qquad \Delta W = \frac{\alpha}{r} BA, \tag{3}$$

where $A \in \mathbb{R}^{r\times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}}\times r}$ are learned low-rank factors, r is the LoRA rank, and α/r is the update scale. The collection of layer-wise updates forms $\Delta\theta$, yielding the adapted VLM $M_1 = f_{\theta_0+\Delta\theta}$. After fine-tuning, the victim releases the LoRA adapter but keeps $\mathcal{D}_{\text{priv}}$ private.

Attacker's goal. The attacker is given a target condition $c_t \in \mathcal{C}$, a public task query q_t, and a public image generator comprising a mapping network $G_{\text{map}} : \mathcal{Z} \to \mathcal{W}$ and a synthesis network $G_{\text{syn}} : \mathcal{W} \to \mathcal{X}$. The goal is to recover visual content associated with c_t in the private fine-tuning data. At a high level, the attacker searches for

$$w_t^\star = \arg\min_{w\in\mathcal{W}} \mathcal{J}_{\text{inv}}\left(G_{\text{syn}}(w); c_t, q_t, M_0, M_1\right), \qquad \widehat{x}_t = G_{\text{syn}}(w_t^\star). \tag{4}$$

Here, $\mathcal{J}_{\text{inv}}$ denotes the overall inversion criterion. As detailed in Section 4, VisDelta refines each candidate with adapter-aware differential guidance and a latent-space constraint, then selects the reconstruction whose target responses are most consistent across freshly

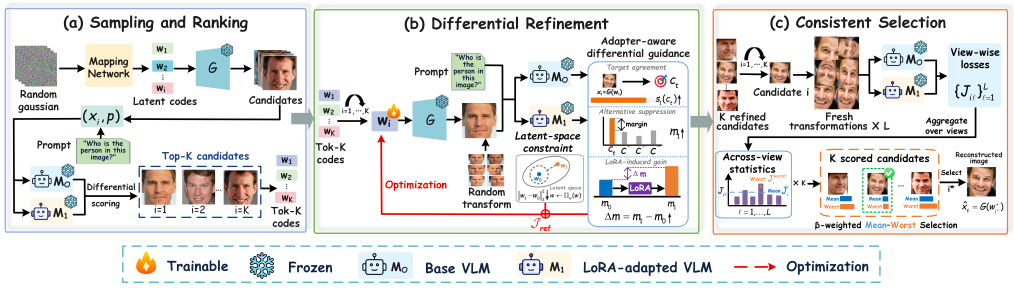

Figure 2: **Overview of VisDelta. (a)** Candidate images sampled from a public generator are ranked by differential scores computed from the paired responses of the base VLM M_0 and the LoRA-adapted VLM M_1. The top K latent codes are retained. **(b)** Each retained code is independently refined under random image transformations using adapter-aware differential guidance, which combines target agreement, suppression of alternative targets, LoRA-induced gain, and a latent-space constraint. **(c)** The refined candidates are reevaluated over L fresh transformations and ranked using a weighted combination of their mean and worst-view guidance losses. Both VLMs and generator G remain frozen; only the latent codes are optimized. Faces are shown for illustration.

Attacker's knowledge and capabilities. The attacker has white-box access to the public base VLM M_0 and released LoRA adapter, which instantiate M_1. The target c_t, complete condition set $\mathcal{C}$, task query template, and public image generator are known. For generated images, the attacker can obtain token likelihoods from both VLMs and backpropagate through them and the generator to optimize latent codes. The attacker cannot access $\mathcal{D}_{\text{priv}}$, intermediate checkpoints, optimizer states, or victim-training gradients. Only latent codes are updated during inversion; all VLM and generator parameters remain frozen throughout the attack.

4 Method

4.1 Overview

In this paper, we propose **VisDelta**, a model inversion attack that reconstructs target-related visual content from released VLM LoRA adapters. Let M_0 and M_1 denote the public base and LoRA-adapted VLMs, respectively. Since a high target response under M_1 may reflect knowledge already present in M_0, VisDelta evaluates each generated image and query with both models. Their difference in target margin isolates the increase in target separation induced by LoRA and serves as the reconstruction signal. As shown in Figure 2, VisDelta comprises three stages. **Sampling and ranking** draws N latent codes from a public generator G and retains the K candidates with the lowest guidance losses. **Differential refinement** optimizes each retained code under random transformations using target agreement, hard-negative separation, and the LoRA-induced differential margin. A latent-space constraint combines a prior penalty with projection to limit deviation from the initialization. **Consistent selection** reevaluates the refined candidates under L fresh transformations and selects the one minimizing a weighted combination of the mean and worst-view guidance losses. All VLM and generator parameters remain frozen; only the latent codes are optimized.

4.2 Adapter-Aware Differential Guidance

For a generated image x, VisDelta evaluates the same condition and query under the base and adapted VLMs. Comparing their paired responses distinguishes adapter-induced changes from support already present in the base model. These quantities define the guidance used for candidate ranking, latent refinement, and consistent selection.

Condition score and response gain. For query q_t, let $c = (c_1, \ldots, c_{T_c})$ denote an answer sequence of T_c tokens representing a condition. For model M_j, where $j = 0$ and $j = 1$ denote the base and adapted VLMs, respectively, we define:

$$s_j(x, c; q_t) = \frac{1}{T_c} \sum_{r=1}^{T_c} \log p_{M_j}\left(c_r \mid c_{<r}, x, q_t\right). \tag{5}$$

Here, c_r is the r-th token, $c_{<r} = (c_1, \ldots, c_{r-1})$ is its preceding prefix, and p_{M_j} denotes the conditional next-token probability. Conditioning on the known prefix yields teacher-forced scoring, while division by T_c normalizes for answer length. For $T_c = 1$, s_j reduces to the log-probability of the single answer token.

Target separation and differential margin. Let $\mathcal{H}_t$ be the fixed set of $H = |\mathcal{H}_t|$ hard-negative conditions selected during sampling and ranking in Section 4.3. Each $c^- \in \mathcal{H}_t$ denotes a condition competing with the target c_t. Under model M_j, the target margin is:

$$m_j(x, c_t; q_t) = s_j(x, c_t; q_t) - \frac{1}{H} \sum_{c^- \in \mathcal{H}_t} s_j(x, c^-; q_t). \tag{6}$$

Thus, m_j compares the target score with the mean hard-negative score under M_j, with larger values indicating stronger target separation. The LoRA-induced margin change is:

$$\Delta m(x, c_t; q_t) = m_1(x, c_t; q_t) - m_0(x, c_t; q_t). \tag{7}$$

Here, m_0 and m_1 are the target margins under the base and adapted VLMs, respectively. The expanded form expresses Δm as the target response gain minus the mean gain of its hard negatives, discounting both confidence inherited from M_0 and adapter effects shared across conditions. Consequently, VisDelta defines its differential loss on Δm, rather than on the raw response gain g_Δ.

Differential guidance construction. The definitions above provide three guidance signals: target agreement under M_1, separation from hard negatives, and the increase in target separation induced by LoRA. For the two margin-based terms, we use $\rho(u) = \tanh(u/\tau)$, where $\tau > 0$ controls saturation and prevents already large margins from dominating latent optimization.

Target agreement. We enforce target agreement under M_1 using:

$$\mathcal{L}_{\text{answer}}(x) = -s_1(x, c_t; q_t). \tag{8}$$

The answer term raises the length-normalized target score under M_1.

Alternative suppression. Target support alone is insufficient when competing conditions receive similar scores. We encourage target separation using:

$$\mathcal{L}_{\text{margin}}(x) = -\rho(m_1(x, c_t; q_t)). \tag{9}$$

The margin term widens the gap between the target and mean hard-negative scores under M_1.

LoRA-induced separation gain. Because a large margin under M_1 may be inherited from the base model, we isolate the separation introduced by LoRA:

$$\mathcal{L}_\Delta(x) = -\rho(\Delta m(x, c_t; q_t)). \tag{10}$$

The differential term favors images whose target separation increases after loading the adapter.

Adapter-aware guidance objective. The complete guidance objective is:

$$\mathcal{J}_{\text{guide}}(x; c_t, q_t) = \lambda_{\text{ans}} \mathcal{L}_{\text{answer}}(x) + \lambda_{\text{mar}} \mathcal{L}_{\text{margin}}(x) + \lambda_\Delta \mathcal{L}_\Delta(x), \tag{11}$$

where λ_{ans}, λ_{mar}, and λ_Δ are nonnegative weights. Minimizing $\mathcal{J}_{\text{guide}}$ jointly promotes target agreement, hard-negative separation, and adapter-induced separation gain. The same objective is used for candidate ranking, latent refinement, and view-wise scoring during consistent selection.

4.3 Sampling and Ranking

VisDelta samples N Gaussian vectors and maps them through the public generator:

$$z_n \sim \mathcal{N}(0, I), \qquad w_n = G_{\text{map}}(z_n), \qquad x_n = G_{\text{syn}}(w_n), \quad n = 1, \ldots, N. \tag{12}$$

Here, I is the identity covariance matrix. The resulting pool $\mathcal{P} = \{(w_n, x_n)\}_{n=1}^N$ supports both hard-negative mining and candidate ranking.

For each non-target condition $c \in \mathcal{C}$, we measure its competition with c_t across the candidate pool:

$$h_t(c) = \frac{1}{N} \sum_{n=1}^{N} \sigma(s_1(x_n, c; q_t) - s_1(x_n, c_t; q_t)), \tag{13}$$

where σ is the sigmoid function. A larger $h_t(c)$ indicates that c more consistently competes with the target under M_1. The H conditions with the largest scores form $\mathcal{H}_t$, which remains fixed in all subsequent stages.

Finally, each candidate is assigned $R_n = \mathcal{J}_{\text{guide}}(x_n; c_t, q_t)$. VisDelta retains the K latent codes with the lowest ranking losses and reindexes them as $\{w_{i,0}\}_{i=1}^{K}$ for differential refinement.

4.4 Differential Refinement

Each retained code is independently optimized under differentiable image transformations, while a latent-space constraint limits its deviation from the selected initialization.

Latent-space constraint. For initialization $w_{i,0}$, the feasible search region is

$$\mathcal{B}_i = \{w \in \mathcal{W} : \|w - w_{i,0}\|_2 \le \epsilon\}, \tag{14}$$

where ϵ is the search radius. For transformation a, the per-view loss is

$$\mathcal{J}_{\text{ref}}(w, a; i) = \mathcal{J}_{\text{guide}}\left(a(G_{\text{syn}}(w)); c_t, q_t\right) + \lambda_{\text{prior}}\|w - w_{i,0}\|_2^2, \tag{15}$$

where $\lambda_{\text{prior}} \ge 0$ controls the quadratic prior. The prior penalty limits drift within $\mathcal{B}_i$, while projection onto $\mathcal{B}_i$ enforces the hard search radius.

Transformation-aware optimization. We minimize its expectation under $\mathcal{A}_{\text{opt}}$:

$$w_i^\star = \arg\min_{w \in \mathcal{B}_i} \mathbb{E}_{a \sim \mathcal{A}_{\text{opt}}}\left[\mathcal{J}_{\text{ref}}(w, a; i)\right], \tag{16}$$

where $\mathcal{A}_{\text{opt}}$ is the optimization transformation distribution. Starting from $w_{i,0}$, VisDelta performs T projected Adam steps with a fresh transformation per step. Both VLMs evaluate the same view to compute matched-input differential margins. Gradients pass through the frozen VLMs and generator to update only w, which is projected onto $\mathcal{B}_i$. We use random resized crops, translations, horizontal flips, and brightness and contrast jitter (Athalye et al., 2018; Struppek et al., 2022).

4.5 Consistent Selection

Refinement produces K latent codes $\{w_i^\star\}_{i=1}^{K}$. To evaluate their stability beyond the views used for optimization, VisDelta samples L fresh transformations for each candidate:

$$a_{i\ell} \overset{\text{i.i.d.}}{\sim} \mathcal{A}_{\text{sel}}, \qquad i = 1, \dots, K, \quad \ell = 1, \dots, L. \tag{17}$$

The distribution $\mathcal{A}_{\text{sel}}$ uses the same transformation family as $\mathcal{A}_{\text{opt}}$, with parameters sampled independently of those used during refinement. The loss under view ℓ is

$$J_{i\ell} = \mathcal{J}_{\text{guide}}\left(a_{i\ell}(G_{\text{syn}}(w_i^\star)); c_t, q_t\right). \tag{18}$$

For each candidate, we compute the mean and worst-view losses:

$$\overline{J}_i = \frac{1}{L}\sum_{\ell=1}^{L} J_{i\ell}, \qquad J_i^{\text{worst}} = \max_{1 \le \ell \le L} J_{i\ell}. \tag{19}$$

The mean captures performance across typical views, while the maximum penalizes failure under the most difficult transformation. Their weighted combination defines the selection loss:

$$J_i^{\text{sel}} = (1 - \beta)\overline{J}_i + \beta J_i^{\text{worst}}, \qquad \beta \in [0, 1], \tag{20}$$

where β controls the worst-view weight. The final output is

$$i^\star = \arg\min_{1 \le i \le K} J_i^{\text{sel}}, \qquad \widehat{x}_t = G_{\text{syn}}(w_{i^\star}^\star). \tag{21}$$

Selection uses $\mathcal{J}_{\text{guide}}$ rather than $\mathcal{J}_{\text{ref}}$, because the latent prior constrains refinement but should not affect comparison among the resulting candidates. This stage performs no gradient updates.

Table 1: Main inversion results on Qwen2.5-VL-7B. Best and second-best results within each dataset are bolded and underlined, respectively.

Dataset	Method	Recovery		Feature distance		Visual measures		
		Acc@1↑	Acc@5↑	δ_{face} ↓	δ_{eval} ↓	FID↓	CS↑	LPIPS↓
FaceScrub	Ranking Only	7.42%	12.18%	1.218	174.365	138.42	0.372	0.812
	Final CE Only	19.36%	44.88%	0.936	134.572	118.74	0.578	0.721
	Base Only	11.63%	34.27%	1.082	157.918	127.36	0.497	0.784
	SMI-AW	<u>21.72%</u>	<u>48.63%</u>	<u>0.907</u>	<u>131.218</u>	**109.86**	<u>0.613</u>	<u>0.694</u>
	VisDelta (Ours)	**25.14%**	**51.26%**	**0.842**	**121.683**	<u>112.31</u>	**0.641**	**0.686**
CelebA	Ranking Only	8.17%	13.44%	1.238	205.371	143.87	0.351	0.801
	Final CE Only	20.84%	46.93%	0.968	161.742	115.26	0.607	0.735
	Base Only	13.29%	35.74%	1.114	183.256	132.48	0.532	0.764
	SMI-AW	<u>22.57%</u>	<u>50.18%</u>	<u>0.931</u>	<u>157.864</u>	**109.74**	<u>0.629</u>	**0.703**
	VisDelta (Ours)	**26.38%**	**52.47%**	**0.886**	**149.385**	<u>111.92</u>	**0.648**	<u>0.711</u>

5 Experiments

5.1 Experimental Setup

Models and datasets. We evaluate VisDelta on Qwen2.5-VL-7B (Wu C, 2025), LLaVA-v1.6-Mistral-7B (Liu et al., 2024), MiniCPM-V-2.6 (Yao et al., 2024), and InternVL3-8B (Zhu et al., 2025). Experiments use FaceScrub (Ng & Winkler, 2014), the 200 CelebA identities with the most images (Liu et al., 2015), and all 120 Stanford Dogs breeds (Khosla et al., 2011).

Compared methods. We compare VisDelta with Ranking Only, Base Only, Final CE Only, and SMI-AW. Ranking Only selects the highest-ranked generator sample without refinement; Base Only optimizes against M_0; and Final CE Only minimizes target cross-entropy under M_1. SMI-AW (Nguyen et al., 2026) uses adaptive token weighting and logit maximization. All methods share the same splits, queries, labels, candidate budgets, and evaluation settings.

Implementation details. For each target, VisDelta samples $N = 512$ latent codes and independently refines the top $K = 8$ for $T = 50$ Adam steps using $H = 4$ hard negatives. It then returns the candidate with the lowest selection loss over $L = 10$ fresh transformations. We use FFHQ-trained StyleGAN2 for faces and StarGAN v2 trained on AFHQ-Dog for dogs (Karras et al., 2020; Choi et al., 2020). More details in Appendix B.

Metrics. We report Acc@1/5, δ_{face}, δ_{eval}, FID, CS, and LPIPS. Acc@K is target top-K accuracy from an independent Inception-v3 evaluator (Szegedy et al., 2016). For faces, δ_{face} measures reconstruction distance in the FaceNet space (Schroff et al., 2015), while δ_{eval} uses the evaluator's feature space. FID measures distributional discrepancy (Heusel et al., 2017), CS is OpenCLIP cosine similarity (Cherti et al., 2023), and LPIPS measures perceptual distance (Zhang et al., 2018). Additional details and complete metric definitions are provided in Appendix C.

5.2 Main Results

Main inversion performance. Table 1 shows that VisDelta achieves the best Acc@1/5, δ_{face}, δ_{eval}, and CS on both datasets. It outperforms SMI-AW by 3.42/2.63 percentage points on FaceScrub and 3.81/2.29 points on CelebA in Acc@1/5. VisDelta also achieves the best LPIPS on FaceScrub and ranks second in FID on both datasets and LPIPS on CelebA. The gains over Base Only and Final CE Only show that contrasting M_0 and M_1 isolates adapter-specific reconstruction signals more effectively than optimizing either model alone. Representative results are shown in Figure 3.

Generalization across VLMs and domains. Table 2 shows that VisDelta remains effective across model architectures and visual domains. On FaceScrub, Acc@1 ranges from 19.86% to 25.14% and Acc@5 from 44.12% to 51.26% across four VLMs, with Qwen2.5-VL-7B achieving the highest accuracy and InternVL3-8B the lowest FID. VisDelta also transfers from face identities to fine-grained dog breeds. On Stanford Dogs, Qwen2.5-VL-7B obtains higher accuracy and lower δ_{eval}, FID, and LPIPS, while LLaVA-1.6-7B achieves higher CS. Together, the results establish generalization beyond a single architecture or face domain, with representative reconstructions shown in Figure 4.



Figure 3: Representative reconstructions on FaceScrub using Qwen2.5-VL-7B. **Additional examples are provided in the appendix F.1.**



Figure 4: Representative Stanford Dogs reconstructions using LLaVA-1.6-7B. **Additional examples are provided in the appendix F.1.**

Table 2: Generalization of VisDelta across VLM architectures and visual domains. Qwen2.5-VL-7B rows are highlighted in light blue. Best and second-best results within each dataset are bolded and underlined, respectively. δ_{face} is not applicable to Stanford Dogs.

Dataset	Model	Recovery		Feature distance		Visual measures		
		Acc@1↑	Acc@5↑	δ_{face} ↓	δ_{eval} ↓	FID↓	CS↑	LPIPS↓
FaceScrub	LLaVA-1.6-7B	22.47%	47.39%	0.874	129.438	119.84	0.612	0.718
	InternVL3-8B	23.58%	48.76%	0.861	126.715	**110.94**	0.625	0.701
	MiniCPM-V-2.6	19.86%	44.12%	0.893	132.547	124.36	0.596	0.736
	Qwen2.5-VL-7B	**25.14%**	**51.26%**	**0.842**	**121.683**	112.31	**0.641**	**0.686**
CelebA	LLaVA-1.6-7B	20.91%	45.28%	0.915	157.284	118.67	0.603	0.742
	Qwen2.5-VL-7B	**26.38%**	**52.47%**	**0.886**	**149.385**	**111.92**	**0.648**	**0.711**
Stanford Dogs	LLaVA-1.6-7B	30.84%	63.63%	–	118.952	128.84	**0.641**	0.792
	Qwen2.5-VL-7B	**32.73%**	**66.92%**	–	**111.604**	**121.63**	0.592	**0.768**

Component sensitivity and objective trade-offs. Figure 5(c) shows the largest degradation after removing candidate ranking or hard-negative suppression, confirming their roles in initialization quality and target discrimination. Transformation-aware refinement and consistent selection provide additional gains by reducing sensitivity to individual views. As shown in Figure 5(d), the complete objective achieves the best FID–Acc@1 trade-off. The answer and margin terms support target recovery, the differential term improves the trade-off, and the latent prior preserves visual quality.

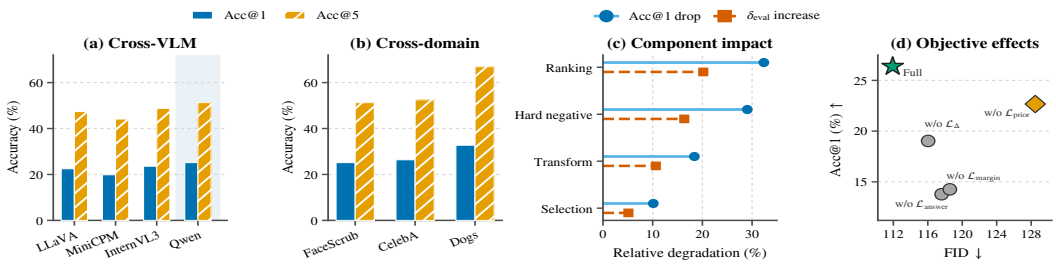

Figure 5: Generalization and diagnostic analysis of VisDelta. **(a)** Cross-model recovery accuracy on FaceScrub. **(b)** Cross-domain recovery accuracy using Qwen2.5-VL-7B. **(c)** Sensitivity to individual pipeline components, measured by the relative degradation from the complete configuration. **(d)** FID–Acc@1 trade-off under different objective configurations, where the upper-left region is preferred.

5.3 Ablation Study

We evaluate VisDelta on CelebA with Qwen2.5-VL-7B to isolate the contributions of its pipeline stages, guidance terms, and latent-space constraint. Each ablation removes only the indicated component while keeping the remaining settings fixed. We further retain the complete attack procedure but use a wrong adapter or permuted condition to test whether recovery depends on the intended adapter and target condition. Table 3 summarizes the results.

Removing candidate ranking or hard-negative mining produces the largest drops among pipeline ablations, while transformation-aware refinement and consistent selection provide complementary gains. Target agreement and alternative suppression are the most influential guidance terms; removing

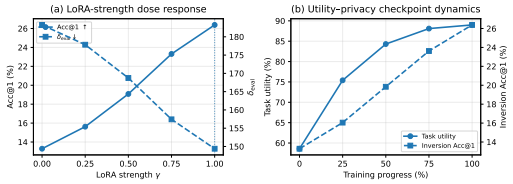

Figure 6: Leakage dynamics under adapter scaling and training checkpoints. Stronger LoRA adaptation increases inversion performance.

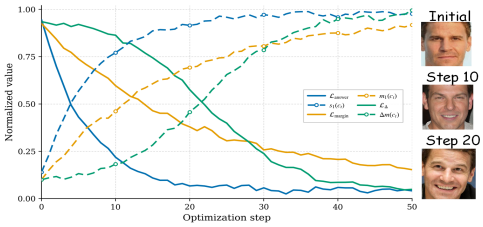

Figure 7: Optimization dynamics over 50 latent refinement steps.

Table 3: Ablation and control results on CelebA with Qwen2.5-VL-7B. Each variant changes one setting; the complete VisDelta configuration is highlighted in light blue.

Configuration	Recovery		Feature distance		Visual measures		
	Acc@1↑	Acc@5↑	$\delta_{\text{face}}\downarrow$	$\delta_{\text{eval}}\downarrow$	FID↓	CS↑	LPIPS↓
(a) Pipeline components							
w/o candidate ranking	17.84%	39.63%	1.031	179.527	119.64	0.574	0.740
w/o hard-negative mining	18.72%	41.26%	0.998	173.814	117.17	0.591	0.735
w/o transformation-aware refinement	21.53%	45.42%	0.954	165.276	115.41	0.609	0.727
w/o consistent selection	23.71%	49.38%	0.918	157.036	113.27	0.630	0.719
(b) Guidance and latent-space constraint							
w/o $\mathcal{L}_{\text{answer}}$	13.78%	32.91%	1.094	198.732	117.61	0.514	0.765
w/o $\mathcal{L}_{\text{margin}}$	14.26%	34.57%	1.161	190.842	118.56	0.529	0.758
w/o $\mathcal{L}_{\Delta}$	19.03%	42.15%	1.002	171.456	116.04	0.589	0.733
w/o latent-space constraint	22.67%	47.89%	0.931	160.328	128.48	0.617	0.747
(c) Control settings							
wrong adapter	6.24%	22.81%	1.307	226.918	145.37	0.319	0.840
permuted condition	7.36%	24.57%	1.276	217.604	141.11	0.343	0.825
VisDelta (full)	**26.38%**	**52.47%**	**0.886**	**149.385**	**111.92**	**0.648**	**0.711**

$\mathcal{L}_{\Delta}$ still lowers Acc@1 by 7.35 percentage points. The latent-space constraint primarily preserves visual quality, reflected by FID and LPIPS. Wrong-adapter and permuted-condition controls sharply reduce recovery, confirming that recovery depends on the matched adapter and target condition.

5.4 More Discussions

Leakage dynamics during adaptation. Figure 6 shows that increasing the adapter scale raises inversion accuracy and reduces feature distance, indicating that stronger LoRA updates expose a clearer reconstruction signal. Across training checkpoints, leakage continues to increase even as task utility approaches saturation, suggesting that recoverable visual traces can accumulate after utility gains diminish. Together, these trends link recovery to information learned during LoRA adaptation rather than knowledge already present in the base model.

Optimization dynamics. Figure 7 shows that the answer term stabilizes first, followed by the margin and differential terms. This ordering indicates that early updates establish target agreement, whereas later updates suppress competing conditions and amplify LoRA-induced separation. Reconstructions become progressively more target-specific as the losses stabilize, matching the progression from target agreement to adapter-specific discrimination rather than generic target confidence alone.

6 Conclusion

In this paper, we presented **VisDelta**, a model inversion attack that reconstructs target-related visual content from released VLM LoRA adapters without accessing private images. VisDelta isolates adapter-induced information by contrasting target separation under the base and adapted VLMs, then combines differential guidance, latent-space-constrained refinement, and transformation-consistent selection. Experiments on four VLMs and three benchmarks show consistent recovery gains, revealing that released LoRA adapters can leak visual information from private fine-tuning data.

AI USE STATEMENT

Generative AI tools were used to assist with English translation and the drafting and editing of portions of this manuscript. Their use was limited to sentence-level rephrasing, grammar correction, readability improvement, and local organization of technical content provided by the authors. The authors determined the research questions, methodology, experimental protocol, interpretation of results, and conclusions, and independently verified all equations, citations, numerical values, and technical claims against the implementation and experimental records. Generative AI tools were not used to generate datasets or experimental results or to execute experiments. All AI-assisted text was reviewed and revised by the authors. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work investigates privacy leakage from released VLM LoRA adapters and therefore has potential dual-use implications. VisDelta is presented as a privacy-auditing method for characterizing such leakage and motivating safer adapter release, rather than as a tool for accessing data without authorization. The experiments use established public benchmarks and publicly available model and generator checkpoints in controlled research settings; no new human-subject data or nonpublic third-party data were collected. Benchmark images are treated as protected fine-tuning data solely to simulate the threat model, and no third-party private adapters are attacked. Any application of the proposed attack should be restricted to models and data that the evaluator is authorized to examine and should comply with applicable dataset licenses, privacy regulations, and institutional policies. The findings motivate pre-release leakage auditing and the development of privacy-aware fine-tuning and adapter-distribution mechanisms.

REPRODUCIBILITY STATEMENT

Section 3 specifies the threat model and attacker capabilities, while Section 4 provides the condition scores, guidance objectives, latent constraints, and three-stage reconstruction procedure. Section 5 describes the evaluated VLMs, datasets, baselines, public generators, metrics, and principal experimental settings and discussions. Appendix B and C provide the data splits, task queries, preprocessing procedures, adapter-training and inversion hyperparameters, transformation ranges, evaluator configurations, complete metric definitions, and additional results. The appendix also includes pseudocode for the complete attack. All external datasets, model checkpoints, and image generators used in the experiments are publicly available and cited in the paper.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, and Miech. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Shengwei An, Guanhong Tao, Qiuling Xu, Yingqi Liu, Guangyu Shen, Yuan Yao, Jingwei Xu, and Xiangyu Zhang. Mirror: Model inversion for deep learning network with high fidelity. In *Proceedings of the 29th Network and Distributed System Security Symposium*. Internet Society, 2022.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293. PMLR, 2018.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, pp. 2633–2650. USENIX Association, 2021.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium*, pp. 5253–5270. USENIX Association, 2023.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023. doi: 10.1109/CVPR52729.2023.00276.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8188–8197, 2020.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023. URL https://openreview.net/forum?id=OUIFPHEgJU.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333. ACM, 2015. doi: 10.1145/2810103.2813677.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132:581–595, 2024. doi: 10.1007/s11263-023-01891-x.

Lukas Helff, Felix Friedrich, Manuel Brack, Kristian Kersting, and Patrick Schramowski. LlavaGuard: An open VLM-based framework for safeguarding vision datasets and models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22964–22987. PMLR, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yuke Hu, Zheng Li, Zhihao Liu, Yang Zhang, Zhan Qin, Kui Ren, and Chun Chen. Membership inference attacks against large vision-language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 1589–1608, Seattle, WA, August 2025. USENIX Association. ISBN 978-1-939133-52-6. URL https://www.usenix.org/conference/usenixsecurity25/presentation/hu-yuke.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19113–19122, 2023.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28541–28564, 2023a. doi: 10.52202/075280-1240.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023b.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1516.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge. Official LLaVA project page, January 2024.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015.

Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. Cheap and quick: Efficient vision-language instruction tuning for large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 29615–29627, 2023.

Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *IEEE International Conference on Image Processing*, pp. 343–347, 2014. doi: 10.1109/ICIP.2014.7025068.

Ngoc-Bao Nguyen, Sy-Tuyen Ho, Koh Jun Hao, and Ngai-Man Cheung. Do vision-language models leak what they learn? adaptive token-weighted model inversion attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10283–10292, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Delong Ran, Xinlei He, Tianshuo Cong, Anyu Wang, Qi Li, and Xiaoyun Wang. Lora-leak: Membership inference attacks against LoRA fine-tuned language models. *arXiv preprint arXiv:2507.18302*, 2025. doi: 10.48550/arXiv.2507.18302. URL `https://arxiv.org/abs/2507.18302`.

Hasin Us Sami, Swapneel Sen, Amit K. Roy-Chowdhury, Srikanth V. Krishnamurthy, and Basak Güler. Gradient inversion attacks on parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10224–10234, 2025.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015. doi: 10.1109/CVPR.2015.7298682.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *Computer Vision – ECCV 2024*, volume 15110 of *Lecture Notes in Computer Science*, pp. 256–274. Springer, 2024. doi: 10.1007/978-3-031-72943-0_15.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6048–6058, 2023.

Lukas Struppek, Dominik Hintersdorf, Antonio De Almeida Correia, Antonia Adler, and Kristian Kersting. Plug & play attacks: Towards robust and flexible model inversion attacks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20522–20545. PMLR, 2022.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5227–5237, 2022.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Kuan-Chieh Wang, Yan Fu, Ke Li, Ashish Khisti, Richard S. Zemel, and Alireza Makhzani. Variational model inversion attacks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9706–9719, 2021.

Yixu Wang, Yan Teng, Yingchun Wang, and Xingjun Ma. Stolenlora: Exploring LoRA extraction attacks via synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 263–272, 2025.

Zhou J Wu C, Li J. Qwen2.5-VL technical report, 2025.

Dixi Yao. Risks when sharing lora fine-tuned diffusion model weights. *arXiv preprint arXiv:2409.08482*, 2024. doi: 10.48550/arXiv.2409.08482. URL `https://arxiv.org/abs/2409.08482`.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm-v: A gpt-4v level mllm on your phone, 2024.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16337–16346, 2021.

Xiaojian Yuan, Kejiang Chen, Jie Zhang, Weiming Zhang, Nenghai Yu, and Yang Zhang. Pseudo label-guided model inversion attack via conditional generative adversarial network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3):3349–3357, 2023. doi: 10.1609/aaai.v37i3.25442.

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. doi: 10.1109/CVPR.2018.00068.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 253–261, 2020.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022a. doi: 10.1109/CVPR52688.2022.01631.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022b. doi: 10.1007/s11263-022-01653-1.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, and Ye. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A Adapter-Induced Visual Signals and Privacy Leakage

This section traces the differential signal used by VisDelta from the released LoRA update to the image-dependent guidance available during inversion. The derivations use the condition scores and target margins defined in Section 4.

A.1 Target-specific response change

Let $g_\Delta(x, c; q_t) = s_1(x, c; q_t) - s_0(x, c; q_t)$ be the change in the score of condition c after loading the adapter. For a fixed image, query, and hard-negative set $\mathcal{H}_t$, Eqs. equation 6 and equation 7 give:

$$\Delta m(x, c_t; q_t) = g_\Delta(x, c_t; q_t) - \frac{1}{H} \sum_{c^- \in \mathcal{H}_t} g_\Delta(x, c^-; q_t). \tag{22}$$

The comparison removes changes shared by the target and its alternatives. For example, an image-dependent score shift that is identical for every condition in $\{c_t\} \cup \mathcal{H}_t$ contributes zero to Δm, even if it increases $s_1(x, c_t; q_t)$. A positive Δm means that the adapter has increased the target score relative to the mean hard-negative score. It does not imply that the final margin m_1 is positive: by definition, $m_1 = m_0 + \Delta m$. This distinction motivates the separate target-agreement, final-margin, and differential terms in equation 11.

A.2 How the LoRA update changes the margin

Write $m_\theta(x, c_t; q_t)$ for the margin evaluated at effective weights θ, keeping x, q_t, and $\mathcal{H}_t$ fixed. Let $\theta_u = \theta_0 + u\Delta\theta$. If the margin is absolutely continuous along this path, the fundamental theorem of calculus gives:

$$\Delta m(x, c_t; q_t) = \int_0^1 \nabla_\theta m_{\theta_u}(x, c_t; q_t)^\top \Delta\theta \, \mathrm{d}u. \tag{23}$$

The path interpolates between the base and adapted configurations for analysis; it is not the victim's optimization trajectory. For the set $\mathcal{L}$ of adapted layers, equation 3 expands this identity as:

$$\begin{aligned} \Delta m(x, c_t; q_t) &= \sum_{\ell \in \mathcal{L}} \int_0^1 \langle \nabla_{W_\ell} m_{\theta_u}, \Delta W_\ell \rangle_F \, \mathrm{d}u, \\ \Delta W_\ell &= \frac{\alpha_\ell}{r_\ell} B_\ell A_\ell. \end{aligned} \tag{24}$$

Here m_{θ_u} is evaluated at $(x, c_t; q_t)$, and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product. The margin changes when an update $B_\ell A_\ell$ aligns with the margin's sensitivity at that layer. Low rank limits the form of this update, but does not make the alignment or the resulting response change vanish.

The same expression clarifies adapter-scaling experiments. For $\theta(\gamma) = \theta_0 + \gamma\Delta\theta$, the initial slope is:

$$\frac{\mathrm{d}}{\mathrm{d}\gamma} m_{\theta(\gamma)}(x, c_t; q_t)\Big|_{\gamma=0} = \nabla_\theta m_{\theta_0}(x, c_t; q_t)^\top \Delta\theta. \tag{25}$$

The sign and magnitude can vary across images and conditions. Increasing the adapter scale therefore changes the available signal, but the equation alone does not imply that every target margin rises monotonically.

A.3 Dependence on private fine-tuning examples

The update in equation 23 is learned from $\mathcal{D}_{\mathrm{priv}}$ through equation 2. To identify what constitutes a training-data signal, consider two counterfactual datasets $\mathcal{D}^+$ and $\mathcal{D}^-$ that differ in one target example, with the base model and fine-tuning procedure, including its randomness, held fixed. Let $\Delta\theta^+$ and $\Delta\theta^-$ be their respective effective LoRA updates. Using the same probe image, query, and fixed hard-negative set in both cases, define:

$$\begin{aligned} \Gamma_t(x; c_t, q_t) &= \Delta m_{\mathcal{D}^+}(x, c_t; q_t) - \Delta m_{\mathcal{D}^-}(x, c_t; q_t) \\ &= m_{\theta_0 + \Delta\theta^+} - m_{\theta_0 + \Delta\theta^-}, \end{aligned} \tag{26}$$

where the second line evaluates both margins at $(x, c_t; q_t)$. The base margin cancels exactly. When Γ_t is nonzero, changing that training example changes an observable target-relative response under this probe. The two adapters may nevertheless produce similar responses for other images or queries. This counterfactual comparison defines sample dependence; VisDelta does not require access to $\mathcal{D}^-$ or its adapter. In the actual threat model, the attacker instead compares the available adapted model against the public base model. Membership and image–condition retrieval results test whether the measured differences carry training-related and target-related information, respectively.

Algorithm 1 VisDelta model inversion

Require: Frozen VLMs M_0, M_1 and public generator $(G_{\text{map}}, G_{\text{syn}})$; public context sampler $\mathcal{Q}_G$
Require: Target c_t, query q_t, conditions $\mathcal{C}$; $N, K, H, T, L, \epsilon, \beta, \tau$; $\lambda_{\text{ans}}, \lambda_{\text{mar}}, \lambda_{\Delta}, \lambda_{\text{prior}}$; $\mathcal{A}_{\text{opt}}, \mathcal{A}_{\text{sel}}$
Ensure: Reconstruction $\widehat{x}_t$

for $n = 1, \dots, N$ **do**
  Sample $z_n \sim \mathcal{N}(0, I)$ and $\xi_n \sim \mathcal{Q}_G$
  $w_n \leftarrow G_{\text{map}}(z_n; \xi_n)$
  $x_n \leftarrow G_{\text{syn}}(w_n; \xi_n)$
end for
for all $c \in \mathcal{C} \setminus \{c_t\}$ **do**
  Compute $h_t(c)$ from $\{x_n\}_{n=1}^N$ using equation 13
end for
$\mathcal{H}_t \leftarrow$ the H conditions with largest $h_t(c)$
for $n = 1, \dots, N$ **do**
  $R_n \leftarrow \mathcal{J}_{\text{guide}}(x_n; c_t, q_t)$
end for
Retain $\{(w_{i,0}, \xi_i)\}_{i=1}^K$ corresponding to the K smallest R_n
for $i = 1, \dots, K$ **do**
  $w_i \leftarrow w_{i,0}$; initialize a separate Adam state
  $\mathcal{B}_i \leftarrow \{w \in \mathcal{W} : \|w - w_{i,0}\|_2 \le \epsilon\}$
  for $u = 1, \dots, T$ **do**
    Sample $a \sim \mathcal{A}_{\text{opt}}$
    $x_i \leftarrow G_{\text{syn}}(w_i; \xi_i)$
    $J_i \leftarrow \mathcal{J}_{\text{guide}}(a(x_i); c_t, q_t) + \lambda_{\text{prior}} \|w_i - w_{i,0}\|_2^2$
    $d_i \leftarrow \nabla_{w_i} J_i$ using matched inputs to M_0 and M_1
    $\widetilde{w}_i \leftarrow \text{AdamStep}(w_i, d_i)$
    $w_i \leftarrow \Pi_{\mathcal{B}_i}(\widetilde{w}_i)$
  end for
  $w_i^\star \leftarrow w_i$; $x_i^\star \leftarrow G_{\text{syn}}(w_i^\star; \xi_i)$
end for
for $i = 1, \dots, K$ **do**
  for $\ell = 1, \dots, L$ **do**
    Sample $a_{i\ell} \sim \mathcal{A}_{\text{sel}}$ independently of refinement
    $J_{i\ell} \leftarrow \mathcal{J}_{\text{guide}}(a_{i\ell}(x_i^\star); c_t, q_t)$
  end for
  $\overline{J}_i \leftarrow L^{-1} \sum_{\ell=1}^L J_{i\ell}$
  $J_i^{\text{worst}} \leftarrow \max_{1 \le \ell \le L} J_{i\ell}$
  $J_i^{\text{sel}} \leftarrow (1 - \beta)\overline{J}_i + \beta J_i^{\text{worst}}$
end for
$i^\star \leftarrow \arg\min_{1 \le i \le K} J_i^{\text{sel}}$
return $\widehat{x}_t \leftarrow x_{i^\star}^\star$

Local effect of one training step. The counterfactual definition can be related to the fine-tuning objective without assuming a particular final adapter. Consider a single gradient step from a LoRA initialization $\phi_0 = (A_{\ell,0}, B_{\ell,0})_{\ell \in \mathcal{L}}$ with $B_{\ell,0} = 0$. Let ϕ collect the trainable factors and put $v_i = \nabla_\phi s_{\phi_0}(x_i, c_i; q_i)$. Since the per-example fine-tuning loss in equation 2 is the negative condition score, a gradient step of size η gives $\phi_+ = \phi_0 + (\eta/n) \sum_{i=1}^n v_i$. If $m_\phi(x, c_t; q_t)$ has an L_m-Lipschitz

gradient along this step, Taylor's theorem yields:

$$\begin{aligned} m_{\phi_+}(x, c_t; q_t) - m_0(x, c_t; q_t) &= \frac{\eta}{n} \sum_{i=1}^{n} \kappa_i(x) + R(x), \\ \kappa_i(x) &= \left\langle \nabla_\phi m_{\phi_0}(x, c_t; q_t), v_i \right\rangle, \\ |R(x)| &\le \frac{L_m \eta^2}{2} \left\| \frac{1}{n} \sum_{i=1}^{n} v_i \right\|_2^2 . \end{aligned} \tag{27}$$

For a private example (x_t, c_t, q_t), let $v_c = \nabla_\phi s_{\phi_0}(x_t, c; q_t)$. At its own image,

$$\kappa_t(x_t) = \|v_t\|_2^2 - \frac{1}{H} \sum_{c^- \in \mathcal{H}_t} \langle v_{c^-}, v_t \rangle. \tag{28}$$

The target example raises its differential margin to first order when its score gradient aligns more strongly with itself than with the hard-negative gradients. The other examples and the remainder in equation 27 determine whether this local effect persists in the aggregate update. This one-step calculation explains a possible source of the signal; the exact path identity in equation 23 describes the released adapter after the complete fine-tuning run.

The initial gradients already reflect the low-rank parameterization. For an adapted layer, the chain rule at $B_{\ell,0} = 0$ gives:

$$\begin{aligned} \nabla_{B_\ell} m_{\phi_0}(x, c_t; q_t) &= \frac{\alpha_\ell}{r_\ell} \nabla_{W_\ell} m_0(x, c_t; q_t) A_{\ell,0}^\top, \\ \nabla_{A_\ell} m_{\phi_0}(x, c_t; q_t) &= 0. \end{aligned} \tag{29}$$

The same relation holds for each v_i. Their inner product in equation 27 therefore measures alignment after projection into the LoRA update directions. A small adapter can retain a sample-dependent signal whenever that alignment is nonzero.

A.4 Connection to latent-space reconstruction

An image-dependent differential margin provides a gradient through the frozen public generator. At $x = G_{\mathrm{syn}}(w)$, abbreviate $m_j(x) = m_j(x, c_t; q_t)$, let $\widetilde{\mathcal{L}}_\Delta(w) = \mathcal{L}_\Delta(G_{\mathrm{syn}}(w))$, and write $J_G(w)$ for the generator Jacobian. The chain rule applied to equation 10 gives:

$$\begin{aligned} \nabla_w \widetilde{\mathcal{L}}_\Delta(w) = \; & -\rho'(\Delta m(x, c_t; q_t)) J_G(w)^\top \\ & \cdot \left[\nabla_x m_1(x) - \nabla_x m_0(x) \right]. \end{aligned} \tag{30}$$

Both margins are evaluated on the same generated image; their gradient difference directs latent refinement toward image changes favored by the adapter-induced separation gain. Candidate ranking seeks promising starting points in the generator range, while the final-margin term separately rewards target separation under M_1. The generated image must still represent target-related visual content for this signal to support recovery. That last property is assessed by independent reconstruction metrics and qualitative results, rather than established by the identities above.

B Implementation and Reproducibility Details

B.1 Data partitions and victim adaptation

Experiments use FaceScrub (Ng & Winkler, 2014), the 200 CelebA identities with the most images (Liu et al., 2015), and all 120 Stanford Dogs breeds (Khosla et al., 2011). For each identity or breed, 20 distinct images are assigned to three disjoint subsets: 14 for private LoRA adaptation, three for selecting the adapted checkpoint with the best validation task utility, and three held out for post-hoc evaluation. One held-out image per target is used as the reference for the reported paired reconstruction metrics. The attack does not access any of these images. Validation images are used by the victim for checkpoint selection; the independent reconstruction metrics are computed only after the attack has returned its final image.

LoRA (Hu et al., 2022) is applied to the query, key, value, and output projections of the language model's self-attention layers. The visual encoder, multimodal connector, and other projection modules remain frozen. We use rank $r = 32$, scaling factor $\alpha = 32$, and LoRA dropout 0. The adapter is trained for 10 epochs with AdamW at a learning rate of 10^{-4}, a microbatch size of 1, and 4 gradient-accumulation steps. Weight decay is 0, the maximum gradient norm is 1.0, and the learning rate remains constant after a warmup ratio of 0.02. The base model and selected adapted checkpoint remain fixed during inversion.

B.2 Generator and inversion configuration

Face candidates are produced by an FFHQ-trained StyleGAN2 (Karras et al., 2020); dog candidates use StarGAN v2 trained on AFHQ-Dog (Choi et al., 2020). The generators remain frozen. For each target condition, the attack samples $N = 512$ initial latent codes, renders candidate images, and ranks them using equation 11. The $H = 4$ hard-negative conditions are mined using equation 13 and then fixed. The top $K = 8$ codes are refined independently in the intermediate latent space $\mathcal{W}$. Each code receives $T = 50$ projected Adam updates under equation 15. The two VLMs process the same transformed generated image at each update; gradients update only w. The 50-step budget counts latent updates, rather than generator or VLM forward passes.

After refinement, each of the eight candidates is evaluated under $L = 10$ newly sampled views. The final image minimizes the selection loss in equation 20; no parameters are updated in this stage. Latent refinement uses Adam with learning rate $\eta = 0.01$ and numerical constant $\varepsilon_{\text{Adam}} = 10^{-8}$. After each update, the latent code is projected onto the ℓ_2 region in equation 14, with radius $\epsilon = 1.5$. We use $\lambda_{\text{ans}} = 0.25$, $\lambda_{\text{mar}} = 0.25$, $\lambda_{\Delta} = 1.0$, and $\tau = 1.0$ for guidance. The refinement prior has weight $\lambda_{\text{prior}} = 0.02$, and selection uses $\beta = 0.3$. Neither the private images nor the post-hoc evaluators enter ranking, latent refinement, or final selection.

B.3 Transformation distributions

Each optimization update samples a differentiable transformation, applied identically to the images evaluated by M_0 and M_1. The transformation family comprises a resized crop with scale in $[0.94, 1.00]$, translation of at most 4% of image width and height, horizontal flipping with probability 0.5, a brightness offset in $[-0.035, 0.035]$, and a contrast factor in $[0.94, 1.06]$. Selection uses the same transformation family with draws independent of those used during refinement.

C Evaluation Metrics

For each target $t \in \mathcal{T}$, let c_t be its label, $\widehat{x}_t$ the final reconstruction, and x_t one held-out reference image. Pairwise metrics compare $\widehat{x}_t$ with x_t and are averaged over targets. Evaluators are frozen and used only after the attack. Paired images receive the same RGB preprocessing; face crops use the largest detected face with 1.05 padding, followed by resizing to 224×224.

Target accuracy (Acc@1 and Acc@5). A separate Inception-v3 classifier (Szegedy et al., 2016) predicts the target identity or breed. Let $E(x)$ denote its class scores and $\text{Top}_k(E(x))$ its k highest-scoring labels. Target accuracy is:

$$\text{Acc@}k = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbb{I}[c_t \in \text{Top}_k(E(\widehat{x}_t))], \quad k \in \{1, 5\}. \tag{31}$$

Face feature distance (δ_{face}). A frozen FaceNet encoder (Schroff et al., 2015) provides the face feature $f_{\text{face}}(x)$. For FaceScrub and CelebA, we report the mean Euclidean distance to the held-out face; lower values indicate greater similarity:

$$\delta_{\text{face}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \| f_{\text{face}}(\widehat{x}_t) - f_{\text{face}}(x_t) \|_2 . \tag{32}$$

Evaluator feature distance (δ_{eval}). Let $f_{\text{eval}}(x)$ be the feature extracted by the independent Inception-v3 target evaluator. We average the Euclidean distance between paired features; lower values indicate

closer matches:

$$\delta_{\text{eval}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|f_{\text{eval}}(\widehat{x}_t) - f_{\text{eval}}(x_t)\|_2 \,. \tag{33}$$

Fréchet Inception Distance (FID). FID (Heusel et al., 2017) compares Inception-v3 feature distributions of $\{\widehat{x}_t\}_{t\in\mathcal{T}}$ and $\{x_t\}_{t\in\mathcal{T}}$. Let (μ_g, Σ_g) and (μ_r, Σ_r) be their empirical means and covariances. Smaller FID indicates more similar distributions:

$$\begin{aligned} \text{FID} = \|\mu_g - \mu_r\|_2^2 + \text{Tr}\big(\Sigma_g + \Sigma_r \\ - 2(\Sigma_g^{1/2}\Sigma_r\Sigma_g^{1/2})^{1/2}\big). \end{aligned} \tag{34}$$

OpenCLIP cosine similarity (CS). A frozen OpenCLIP ViT-B/32 encoder (Cherti et al., 2023) extracts image features $h_{\text{CLIP}}(x)$. We average their cosine similarity over targets; higher values indicate better agreement with the held-out image:

$$\text{CS} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{h_{\text{CLIP}}(\widehat{x}_t)^\top h_{\text{CLIP}}(x_t)}{\|h_{\text{CLIP}}(\widehat{x}_t)\|_2 \|h_{\text{CLIP}}(x_t)\|_2}. \tag{35}$$

Learned perceptual distance (LPIPS). LPIPS (Zhang et al., 2018) compares channel-normalized features ψ_ℓ from a frozen AlexNet, using learned channel weights $\mathbf{w}_\ell$. Write $\mathbf{d}_{\ell,h,w}^{(t)} = \psi_\ell(\widehat{x}_t)_{h,w} - \psi_\ell(x_t)_{h,w}$, where H_ℓ, W_ℓ are the layer's spatial dimensions. We average the weighted differences over positions, layers, and targets; lower values indicate closer perceptual appearance:

$$\text{LPIPS} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \sum_{\ell} \frac{1}{H_\ell W_\ell} \sum_{h,w} \left\| \mathbf{w}_\ell \odot \mathbf{d}_{\ell,h,w}^{(t)} \right\|_2^2. \tag{36}$$

D VisDelta Algorithm

The pipeline of proposed VisDelta in Algorithm 1 .

E Additional Experiments

We examine sensitivity to search and refinement budgets, hard-negative mining, and consistency-based selection, followed by experiments on prompt wording and public generators. Unless varied below, the attack samples $N = 512$ initial latent codes, independently refines the top $K = 8$ candidates for $T = 50$ steps, and uses $H = 4$ mined hard negatives. The final reconstruction is selected from $L = 10$ fresh views with $\beta = 0.3$. The latent constraint uses $\epsilon = 1.5$ and $\lambda_{\text{prior}} = 0.02$. The guidance weights are $(\lambda_{\text{ans}}, \lambda_{\text{mar}}, \lambda_\Delta) = (0.25, 0.25, 1.0)$ with $\tau = 1$; the Adam learning rate is $\eta = 0.01$ for latent refinement.

Unless stated otherwise, sensitivity experiments use Qwen2.5-VL-7B on CelebA; the prompt and generator experiments specify their datasets below. The default configuration was fixed before these sensitivity sweeps, which characterize parameter dependence rather than select the reported configuration. Each plotted point or tabulated row represents a measured setting. Relative runtimes, where reported, are normalized to the default configuration.

E.1 Refinement steps

Figure 8(a) shows that Acc@1 rises from 18.42% at $T = 10$ to 26.38% at $T = 50$. Further refinement does not improve Acc@1 in this sweep: it reaches 26.31% at $T = 75$ and 25.96% at $T = 100$, while relative runtime increases from 1.00 at $T = 50$ to 1.44 and 1.87, respectively. These results support the $T = 50$ setting in the main experiments.

E.2 Candidate sampling and refinement budget

The attack ranks N initial codes, refines the top K independently, and returns the candidate with the lowest selection objective J_i^{sel}. At $K = 8$, with all other settings fixed, increasing N from 64

to 512 raises Acc@1 from 18.07% to 26.38% (Figure 8(b)). Increasing N further to 1024 yields 27.01%, a gain of 0.63 percentage points over $N = 512$ at a relative runtime of 1.16. With $N = 512$, increasing K from 1 to 8 raises Acc@1 from 20.31% to 26.38% (Figure 8(c)). At $K = 16$, Acc@1 is 26.24% and relative runtime reaches 1.86. The fixed $N = 512$, $K = 8$ setting offers a favorable accuracy–runtime trade-off over the evaluated budgets.

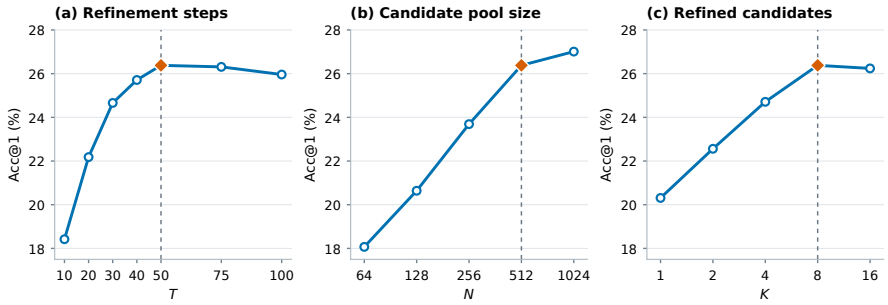

Figure 8: Search and refinement budgets. (a) Latent refinement steps T; (b) initial candidate pool N with $K = 8$ fixed; (c) independently refined candidates K with $N = 512$ fixed. Orange diamonds and dashed vertical lines mark the default settings; three panels share the same Acc@1 vertical scale.

E.3 Hard-negative mining

Table 4(a) compares how the negative conditions are selected. At $H = 4$, mined negatives raise Acc@1 from 18.72% with random selection to 26.38%, a gain of 7.66 percentage points. Acc@5 rises from 41.26% to 52.47%, while δ_{eval} decreases from 173.814 to 149.385. Using every non-target condition gives 25.97% Acc@1 at 1.42 times the default runtime. The four mined negatives therefore achieve the highest measured accuracy in this comparison with a lower cost than using all conditions.

E.4 Consistency-based selection

Figure 9(b) varies the number of fresh selection views at $\beta = 0.3$. Acc@1 rises from 23.28% at $L = 1$ to 26.38% at $L = 10$. Increasing L to 20 yields 26.41%, a further gain of 0.03 percentage points. At $L = 10$, increasing β from 0 to 0.3 raises Acc@1 from 25.06% to 26.38%, whereas $\beta = 1$ yields 25.28% (Figure 9(c)). The default exceeds these mean-only and worst-only rules by 1.32 and 1.10 percentage points, respectively. When $L = 1$, the selection score is independent of β, as reflected by the identical values across the bottom row of the heatmap. The default uses ten views and a moderate worst-view weight.

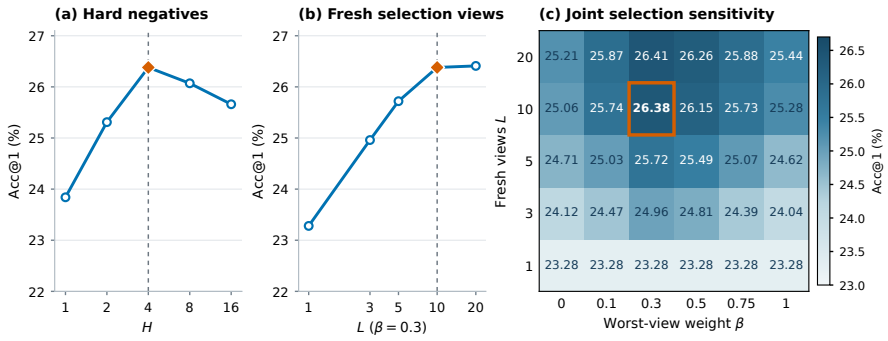

Figure 9: Hard-negative mining and consistency-based selection. (a) Number H of mined hard negatives; (b) number L of fresh selection views at $\beta = 0.3$; (c) joint sweep of L and the worst-view weight β. Heatmap cells report Acc@1 (%). Orange diamonds and the orange heatmap outline mark the default $H = 4, L = 10, \beta = 0.3$ configuration.

E.5 LATENT-SPACE CONSTRAINT

Figure 10(d) compares nine $(\epsilon, \lambda_{\text{prior}})$ configurations. The stronger constraint $(0.75, 0.04)$ yields a FID of 108.91, but reduces Acc@1 to 21.78%. With the looser setting $(3.0, 0.01)$, FID increases to 122.94 and Acc@1 reaches 24.63%. The default $(1.5, 0.02)$ achieves the highest Acc@1 in this grid, at 26.38%, with a FID of 111.92. These results show the trade-off between target recovery and distributional similarity under different latent constraints.

E.6 GUIDANCE AND OPTIMIZATION SENSITIVITY

Figure 10(a–c) varies one parameter at a time around the default. Halving λ_{ans}, λ_{mar}, and λ_{Δ} gives Acc@1 values of 24.72%, 24.28%, and 25.18%, respectively; doubling them gives 25.58%, 25.47%, and 25.83%. The default weights achieve 26.38%. For $\tau = 0.5, 1, 2$, Acc@1 is 25.31%, 26.38%, and 24.91%, respectively. For $\eta = 0.005, 0.01, 0.02$, it is 24.94%, 26.38%, and 25.36%. The default achieves the highest Acc@1 in each one-parameter sweep.

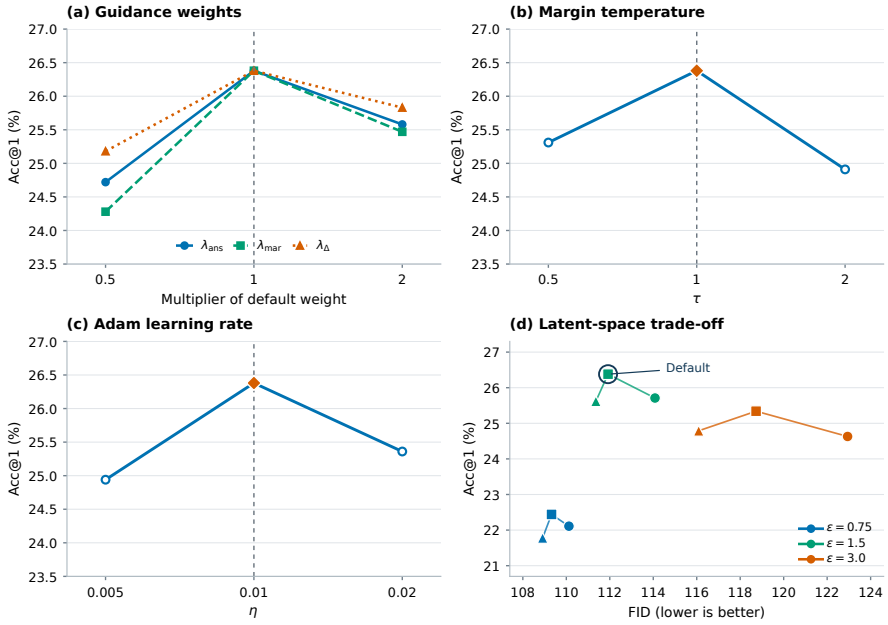

Figure 10: Guidance, optimization, and latent-space sensitivity. (a) Each guidance weight is varied independently by a multiplier of 0.5, 1, or 2; (b) margin temperature τ; (c) Adam learning rate η. The horizontal axes in (a–c) use logarithmic spacing. (d) Acc@1 versus FID for the tested latent radii ϵ and prior weights λ_{prior}. Colors denote $\epsilon = 0.75, 1.5, 3.0$; circles, squares, and triangles denote $\lambda_{\text{prior}} = 0.01, 0.02, 0.04$, respectively. The outlined point marks the default $(\epsilon, \lambda_{\text{prior}}) = (1.5, 0.02)$.

E.7 TRANSFORMATION CHOICES

We examine crop-resize, translation, horizontal flipping, and color jitter, which includes brightness and contrast changes. Their sampling ranges are specified in Appendix B. In Table 4(b), "Refine" denotes transformations sampled during latent-code updates, whereas "Select" denotes fresh views used to compare the fixed refined candidates. A cross means that the corresponding stage uses the untransformed image. The full set includes all four transformation types.

Without transformations, Acc@1 is 20.88%. Applying any one type in both stages raises it to 23.58%–24.41%, while the full set reaches 26.38%. This is 5.50 percentage points above the untransformed setting and 1.97 points above crop-resize, the strongest individual transformation. Using the full set only during refinement gives 23.71% Acc@1, compared with 21.53% when used only during selection. Applying it in both stages gives the highest accuracy and lowers δ_{eval} from 176.204 to 149.385 relative to the untransformed setting.

Table 4: Negative-selection (a) and transformation (b) choices on CelebA with Qwen2.5-VL-7B. The Random and single-stage rows reuse the corresponding ablations in Table 3. All includes every non-target condition; individual transformations are applied in both stages. Full set combines all four types. Acc@1/5 are percentages; runtime is relative to the default.

(a) Negative-selection strategies

Negative selection		Target accuracy		Feature / distribution		Cost
Strategy	H	Acc@1 ↑	Acc@5 ↑	δ_{eval} ↓	FID ↓	Runtime ↓
Random	4	18.72%	41.26%	173.814	117.17	1.15
Mined	4	**26.38%**	**52.47%**	**149.385**	**111.92**	**1.00**
All	All	25.97%	52.31%	150.318	112.15	1.42

(b) Transformation choices

Transformation use			Target accuracy		Feature / distribution	
Setting	Refine	Select	Acc@1 ↑	Acc@5 ↑	δ_{eval} ↓	FID ↓
No transformations	×	×	20.88%	43.91%	176.204	119.92
Crop-resize	✓	✓	24.41%	50.26%	158.552	114.08
Translation	✓	✓	24.16%	49.88%	159.181	114.73
Horizontal flip	✓	✓	23.73%	49.21%	161.044	115.31
Color jitter	✓	✓	23.58%	48.96%	161.721	115.86
Full set: refinement only	✓	×	23.71%	49.38%	157.036	113.27
Full set: selection only	×	✓	21.53%	45.42%	165.276	115.41
Full set: both stages	✓	✓	**26.38%**	**52.47%**	**149.385**	**111.92**

E.8 Prompt robustness

Table 5 compares six question phrasings for each dataset, including the default questions used in the main experiments. Acc@1 ranges from 23.98% to 25.14% on FaceScrub and from 30.96% to 32.73% on Stanford Dogs. The corresponding within-dataset ranges are 1.16 and 1.77 percentage points. The default question gives the highest Acc@1 in both datasets, while another phrasing gives the lowest δ_{eval} on the same set of targets.

Table 5: Prompt robustness on FaceScrub and Stanford Dogs with Qwen2.5-VL-7B. The default questions reproduce the corresponding main-paper results. The default rows are shaded, and the best result for each metric within a dataset is bold.

Input question	Target accuracy		Feature metrics	
	Acc@1 ↑	Acc@5 ↑	δ_{eval} ↓	CS ↑
FaceScrub				
Who is the person in the image?	**25.14%**	**51.26%**	121.683	**0.641**
What is the person's name in the image?	24.21%	50.11%	123.842	0.635
Can you identify the person shown in this photo?	24.89%	50.94%	**120.944**	0.640
Identify the person in this image.	24.56%	50.37%	124.391	0.633
Please identify the person shown here.	24.72%	50.68%	122.918	0.637
The image contains a person. Identify who it is.	23.98%	49.76%	125.207	0.629
Stanford Dogs				
What breed of dog is shown in the image?	**32.73%**	**66.92%**	111.604	0.592
Can you identify the dog breed in this photo?	31.84%	65.88%	113.427	0.585
Please identify the breed of the dog shown here.	32.21%	66.35%	**110.986**	**0.594**
Identify this dog breed.	31.12%	64.93%	115.163	0.580
Given the image, determine the breed of the dog shown.	31.67%	65.74%	112.941	0.588
The image contains a dog. Output its breed.	30.96%	64.58%	114.628	0.582

E.9 PUBLIC GENERATOR PRIOR

Table 6 compares public face generators under the same attack protocol. Among the default StyleGAN2-FFHQ checkpoint, StyleGAN3-FFHQ, and an alternative StyleGAN2-FFHQ checkpoint, Acc@1 ranges from 25.47% to 26.38%; an earlier StyleGAN2-FFHQ checkpoint yields 23.82%. The default generator achieves the highest Acc@1, whereas StyleGAN3-FFHQ gives the lowest FID (109.84) and LPIPS (0.704). With the shifted StyleGAN2-MetFaces prior, Acc@1 falls to 18.42% and FID rises to 138.27. Relative to the default prior, MetFaces lowers Acc@1 by 7.96 percentage points and increases FID by 26.35. The accuracy gap exceeds the 2.56-point range among the four tested FFHQ configurations, indicating greater sensitivity to this domain shift than to the evaluated in-domain generator choices.

Table 6: Sensitivity to the public face generator on CelebA with Qwen2.5-VL-7B. FFHQ generators provide in-domain priors; StyleGAN2-MetFaces represents the shifted prior. The default generator is shaded, and the best value for each metric is bold.

Generator setting		**Target accuracy**		**Feature / distribution**		
Public generator	Domain	Acc@1 ↑	Acc@5 ↑	δ_{eval} ↓	FID ↓	LPIPS ↓
StyleGAN3-FFHQ	In-domain	25.71%	51.92%	151.274	**109.84**	**0.704**
StyleGAN2-FFHQ (alternative checkpoint)	In-domain	25.47%	50.98%	153.418	114.23	0.717
StyleGAN2-FFHQ (earlier checkpoint)	In-domain	23.82%	49.54%	158.762	118.90	0.732
StyleGAN2-MetFaces	Shifted	18.42%	41.37%	176.845	138.27	0.781
StyleGAN2-FFHQ (default)	In-domain	**26.38%**	**52.47%**	**149.385**	111.92	0.711

F ADDITIONAL QUALITATIVE RESULTS

F.1 ADDITIONAL RECONSTRUCTION EXAMPLES

Figures 11–13 provide further examples across two face datasets and Stanford Dogs. In each column, the upper image is a private reference and the lower image is VisDelta's selected reconstruction for the same target condition. The references are shown only for visual comparison and are not used during the attack.



Figure 11: Additional CelebA reconstructions using Qwen2.5-VL-7B. Each column pairs a private reference (top) with the selected reconstruction (bottom) for the same target identity.



Figure 12: Additional FaceScrub reconstructions using Qwen2.5-VL-7B. Each column pairs a private reference (top) with the selected reconstruction (bottom) for the same target identity.



Figure 13: Additional Stanford Dogs reconstructions using LLaVA-v1.6-Mistral-7B. Each column pairs a private reference (top) with the selected reconstruction (bottom) for the same target breed.

G Limitations and Future Works

Limitations. VisDelta assumes access to the base VLM and its released LoRA adapter, a target condition that can be expressed in a query, and a public differentiable generator. The generator defines the images that the attack can search; target-related information in the adapter may therefore remain inaccessible when the generator poorly covers the relevant visual domain. Moreover, the differential margin in Appendix A measures the adapter's response change relative to the selected hard negatives. Its usefulness depends on whether those alternatives expose a distinguishable target signal. The attack also requires repeated generator and VLM evaluations for candidate ranking, latent refinement, and final selection, making computational cost a practical consideration.

Future works. An important direction is to characterize how generator coverage interacts with the information retained by an adapter across broader visual domains. This would help distinguish limitations of the generative prior from limitations of the adapter-induced guidance. Further work could examine inversion under less precise target descriptions and develop more efficient ways to identify promising latent codes. Finally, studying how LoRA rank, module placement, and private training-set composition affect the differential response could clarify which adaptation choices increase or reduce the amount of recoverable visual information.