

Discovering Undisclosed Concepts in LoRA Adapters for Diffusion-Based AIGC Models

Yupeng Qiu¹ Han Fang^{2*} Dhananjai Sharma¹ Ee-Chien Chang¹

¹National University of Singapore

²University of Science and Technology of China

qiu_yupeng@u.nus.edu fanghan@ustc.edu.cn

dhananjaisharma09@gmail.com dcscec@nus.edu.sg

Abstract

Low-Rank Adaptation (LoRA) has made diffusion model personalization affordable and widely shareable, but it also creates moderation risks when harmful or problematic content is not faithfully disclosed. Hence, it is desired to develop a mechanism that, given a LoRA adapter, discovers the concept it was trained to generate, so as to support downstream investigation. We characterize the target concept through the conditioning input that most clearly exposes what the LoRA has learned. Such an input should induce coherent LoRA outputs that are distinctive from those of the base model and strongly align with the learned LoRA subspace. We therefore formulate concept discovery as searching for the conditioning input that jointly maximizes output coherence, LoRA-induced distinctiveness, and subspace alignment. Even with a clear objective, however, optimizing it over an open-ended input space is prohibitively expensive, since reliable evaluation of coherence and distinctiveness requires repeated diffusion rendering. Crucially, we observe that an input need not be rendered to determine whether it is promising. Its alignment with the learned LoRA subspace provides a lightweight signal for localizing promising candidates before visual verification. Based on this insight, we develop a two-stage framework for text-to-image LoRAs. Subspace-Guided Evolutionary Search (SGES) uses render-free subspace alignment to explore candidate prompts and retain a compact shortlist, while Paired Visual Identification (PVI) renders only these candidates and compares the LoRA and base-model outputs to identify those exhibiting coherent and distinctive generation behavior. This design enables efficient and reliable discovery of previously unknown concepts encoded in LoRA adapters.

1 Introduction

Diffusion models have substantially advanced text-to-image (T2I) generation [5, 15, 28, 32, 34–36]. Low-Rank Adaptation (LoRA) [18] further reduces the cost of model customization

and has become widely used for diffusion-model personalization [40, 48]. By learning compact low-rank updates, LoRA can encode diverse visual concepts from only a small set of training images [12, 20]. Its low training and storage costs have facilitated the widespread creation and sharing of customized adapters [18, 48].

However, this accessibility also creates substantial moderation risks [2, 37, 39, 42, 46, 54]. LoRAs can encode copyrighted content, real-person identities, or NSFW concepts, and these capabilities may not be faithfully disclosed by their uploaders [48, 55]. A large-scale study of CivitAI analyzed 221,304 model assets and found that 15.27% (33,804) were flagged as targeting real individuals [48]. Regulatory scrutiny has also increased. For example, the U.S. TAKE IT DOWN Act addresses non-consensual intimate imagery, including AI-generated depictions [45], while the EU AI Act imposes transparency requirements on deepfakes and other AI-generated content [11]. Yet moderation on model-sharing platforms still relies heavily on uploader-provided descriptions, tags, trigger words, and sample images [48]; CivitAI and Hugging Face illustrate the model-sharing setting [4, 19]. Such metadata becomes unreliable when disclosure is incomplete or misleading [55]. Hence, it is desirable to develop a mechanism that, given a LoRA adapter, discovers the concept it was trained to generate.

Existing techniques provide only partial solutions to this problem. Difference-discovery methods identify changes between a customized model and its base model [8, 10, 22], but LoRA adaptation often alters not only the primary training target but also related or collateral content [22]. As a result, detecting what has changed does not reveal which change best represents what the adapter has learned. Concept-verification methods take the opposite route: they test whether a predefined concept is encoded by the LoRA, either through prompt recovery [52] or direct model comparison [55]. Their effectiveness, however, is bounded by the concepts specified in advance. Both directions therefore leave the same fundamental question unresolved: **when the hidden concept is unknown, what exactly should LoRA forensics seek to**

*Corresponding author: Han Fang (fanghan@ustc.edu.cn).

discover?

To answer this question, we characterize the hidden concept through the conditioning input that most clearly exposes what the LoRA has learned. A revealing input should induce a coherent and meaningful output distribution under the LoRA, make this distribution clearly distinctive from that of the base model, and be strongly represented in the learned low-rank subspace. These criteria capture complementary evidence from generation behavior and model parameters: the former establishes whether a stable LoRA-specific behavior is revealed, while the latter indicates whether that behavior is strongly supported by the adapter itself. We therefore formulate hidden-concept discovery as searching for the conditioning input that jointly maximizes output coherence, LoRA-induced distinctiveness, and LoRA-subspace alignment.

However, even with this target well defined, solving the resulting optimization is far from straightforward. The search space of possible conditioning inputs is vast and open-ended, while evaluating output coherence and LoRA-induced distinctiveness requires repeated diffusion rendering. Crucially, we find a strong correspondence between parameter-space alignment and generation-space evidence: inputs that align more strongly with the learned LoRA subspace are substantially more likely to induce coherent and distinctive LoRA-specific behavior after rendering. This finding provides a render-free signal for navigating the large input space, allowing expensive visual evaluation to be concentrated on only a small set of promising candidates.

Building on this finding, we develop a two-stage framework for text-to-image LoRA forensics. We instantiate the conditioning input as a textual prompt and first use **Subspace-Guided Evolutionary Search (SGES)** to explore the open-ended prompt space using LoRA-subspace alignment alone, without diffusion rendering. SGES progressively concentrates the search on highly aligned prompts and produces a compact candidate set. **Paired Visual Identification (PVI)** then renders only these candidates with both the LoRA and base model and identifies those that exhibit coherent LoRA outputs and clear LoRA-induced distinctiveness. In this way, large-scale exploration is performed with a lightweight parameter-space signal, while expensive generation is reserved for final visual identification.

The main contributions of this work are as follows:

- We study the important problem of discovering previously unknown concepts encoded in LoRA adapters and identify three essential characteristics of a valid forensic target: output coherence, LoRA-induced distinctiveness, and LoRA-subspace alignment. These characteristics provide a principled basis for defining and quantifying hidden-concept discovery.
- We uncover a strong correspondence between LoRA-subspace alignment and LoRA-specific generation behavior, and exploit this finding to develop SGES and PVI.

The resulting framework performs large-scale search without diffusion rendering and reserves expensive visual evaluation for only a compact set of candidates.

- We construct a benchmark covering diverse concept types and scenarios, and compare against representative difference-discovery, prompt-recovery, and concept-verification methods. We further introduce a distributional metric to evaluate how faithfully the discovered result captures the hidden content encoded by the LoRA.

2 Background and Related Work

2.1 Diffusion Models

Diffusion models generate images by learning to reverse a gradual stochastic noising process [5, 41]. Denoising Diffusion Probabilistic Models (DDPMs) introduced an effective noise-prediction objective [15], while subsequent work incorporated text conditioning and latent-space denoising, enabling efficient and high-fidelity text-to-image generation [28, 34–36].

Diffusion process. Let $\mathbf{x}_0$ denote a clean image or its latent representation. The forward process gradually adds Gaussian noise according to [15, 41]

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1 - \alpha_t)\mathbf{I}), \quad (1)$$

where $\alpha_t = 1 - \beta_t$ and β_t controls the noise level. Defining $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, a noisy sample at timestep t can be obtained directly as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (2)$$

A denoising network learns to predict the added noise and iteratively reverses this process from Gaussian noise $\mathbf{x}_T$ to $\mathbf{x}_0$. Modern T2I models commonly perform denoising in a compressed latent space and decode the final latent into an image [35].

Text conditioning. For a text input x , a text encoder produces contextual token representations $\mathbf{C}(x) \in \mathbb{R}^{d_c \times L(x)}$, where $L(x)$ is the token length and d_c is the representation dimension. These representations condition the denoising network through cross-attention [35, 47]. Let $\mathbf{F}_t$ denote intermediate visual features at timestep t . The query, key, and value projections are

$$\mathbf{Q} = \mathbf{W}_Q \mathbf{F}_t, \quad \mathbf{K} = \mathbf{W}_K \mathbf{C}(x), \quad \mathbf{V} = \mathbf{W}_V \mathbf{C}(x). \quad (3)$$

The query therefore depends on the evolving visual features, whereas the key and value inputs are determined directly by the text conditioning. This mechanism makes generation sensitive to both the tokens in the text input and their context.

With text conditioning, the denoising network is trained using the noise-prediction objective [15, 35]

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{\mathbf{x}_0, x, t, \boldsymbol{\varepsilon}} \left[\left\| \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t, \mathbf{C}(x)) \right\|_2^2 \right]. \quad (4)$$

At inference time, classifier-free guidance [16] can further strengthen text conditioning by combining conditional and unconditional noise predictions:

$$\hat{\epsilon}_\theta = \epsilon_\theta(\mathbf{x}_t, t, \emptyset) + \eta [\epsilon_\theta(\mathbf{x}_t, t, \mathbf{C}(x)) - \epsilon_\theta(\mathbf{x}_t, t, \emptyset)], \quad (5)$$

where $\emptyset$ denotes the empty condition and η is the guidance scale.

2.2 Low-Rank Adaptation

Parameter-efficient fine-tuning (PEFT) adapts large pretrained models while updating only a small number of parameters [1, 6, 17, 23–25, 53]. Among PEFT methods, Low-Rank Adaptation (LoRA) [18] has become widely used for diffusion-model personalization [40, 48]. LoRA keeps the base model fixed and learns compact low-rank weight updates, enabling visual concepts to be learned from relatively small training sets with low training and storage costs [12, 20].

Low-rank weight adaptation. Consider a pretrained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. LoRA freezes $\mathbf{W}_0$ and represents its learned update as $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{r \times d_{\text{in}}}$, $\mathbf{B} \in \mathbb{R}^{d_{\text{out}} \times r}$, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$ [18]. For an input $\mathbf{z} \in \mathbb{R}^{d_{\text{in}}}$, the adapted layer computes

$$\mathbf{h} = \mathbf{W}_0\mathbf{z} + \Delta\mathbf{W}\mathbf{z}, \quad \Delta\mathbf{W} = \mathbf{B}\mathbf{A}. \quad (6)$$

Only $\mathbf{A}$ and $\mathbf{B}$ are optimized, reducing the number of trainable parameters from $d_{\text{in}}d_{\text{out}}$ to $r(d_{\text{in}} + d_{\text{out}})$.

In T2I diffusion models, LoRA modules can be inserted into selected linear projections in the text encoder and denoising network [20]. The learned updates therefore modify text representations, denoising behavior, or their interactions, depending on the adapted sites. After training, the resulting LoRA weights can be stored and shared independently of the base model [18].

Input-space interpretation. The change introduced by a LoRA update is $\Delta\mathbf{h} = \Delta\mathbf{W}\mathbf{z}$. Let $\Delta\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^\top$ denote its compact singular value decomposition, with rank ρ , right singular vectors $\{\mathbf{v}_j\}_{j=1}^\rho$, and singular values $\{\sigma_j\}_{j=1}^\rho$. The magnitude of the LoRA response satisfies

$$\|\Delta\mathbf{h}\|_2^2 = \|\Delta\mathbf{W}\mathbf{z}\|_2^2 = \sum_{j=1}^\rho \sigma_j^2 (\mathbf{v}_j^\top \mathbf{z})^2. \quad (7)$$

Thus, the response largely depends on the alignment of the input with each right singular vector. Inputs orthogonal to the right singular subspace produce no LoRA response. The leading right singular vectors therefore define the dominant input subspace of the LoRA update, motivating our subsequent parameter-space design.

2.3 Content Moderation for LoRA Adapters

Metadata-based review. Model-sharing platforms primarily rely on uploader-provided descriptions, tags, trigger words,

and sample images to review shared adapters [48]. However, such metadata cannot reliably reveal encoded content when disclosure is incomplete or misleading [55].

Concept-agnostic difference discovery. Related methods discover differences between models or their generated outputs without specifying a concept in advance. VisDiff [10] describes semantic differences between image sets, CompCon [8] searches for prompts that expose differences between T2I models, and DriftScope [22] identifies concept changes caused by model adaptation. Discrete prompt optimization [50] can likewise be adapted to search for LoRA-base differences. However, adaptation may also affect collateral concepts [22]; therefore, identifying what changes does not determine which change best represents the creator-intended concept. Output-based discovery can also require extensive diffusion rendering during search [8, 50].

Concept-specified verification. A complementary direction starts from a concept specified in advance. PEZ [52] recovers prompts for a given concept, providing a way to test whether that concept can be elicited from the LoRA model. PAIA [55] more directly verifies whether a fine-tuned diffusion model has learned a specified concept by comparing its denoising behavior with that of the base model. While such verification can be efficient for an individual concept, its coverage is limited to concepts provided beforehand. Discovering an unknown concept therefore requires repeated verification over a potentially large candidate space.

Summary. Existing methods therefore either discover LoRA-specific changes without identifying the creator-intended one, or verify concepts that must already be specified. In contrast, our work formulates the creator-intended concept as the *dominant hidden concept* and discovers it jointly from LoRA parameter space and generation behavior.

3 Problem Formulation

3.1 Problem Setting

Suppose we are given a LoRA adapter $\mathcal{A}$ and its compatible pretrained T2I base model $\mathcal{B}$. Loading $\mathcal{A}$ into $\mathcal{B}$ yields the LoRA-enabled model $\mathcal{B}_{\mathcal{A}}$. The adapter may affect multiple concepts. We define the *dominant hidden concept* as the creator-intended primary training target that is repeatedly reinforced during adaptation, and distinguish it from collateral concepts affected incidentally [22]. The moderator has access only to $\mathcal{A}$ and $\mathcal{B}$, without reliable metadata, known trigger words, predefined concept queries, reference images, or training data. The goal is to discover a *carrier prompt* p^* that reveals the dominant hidden concept encoded by $\mathcal{A}$.

Prompt-conditioned output distributions. Let $\mathcal{X}$ denote the image space and let $\mathcal{E} : \mathcal{X} \rightarrow \mathbb{R}^q$ be a fixed semantic image encoder [33]. We use its normalized representation

$$\phi(\mathbf{x}) = \frac{\mathcal{E}(\mathbf{x})}{\|\mathcal{E}(\mathbf{x})\|_2}. \quad (8)$$

Let $\mathcal{P}$ denote the prompt space and let $Z \sim \pi_Z$ represent the randomness in diffusion generation. For a prompt $p \in \mathcal{P}$, the semantic output of the LoRA-enabled model satisfies

$$\mathbf{Y}_{\mathcal{A}}(p) = \phi(\mathcal{B}_{\mathcal{A}}(p, Z)) \sim \mathcal{Q}_{\mathcal{A}}(\cdot | p), \quad (9)$$

where $\mathcal{Q}_{\mathcal{A}}(\cdot | p)$ denotes the prompt-conditioned semantic distribution induced by $\mathcal{B}_{\mathcal{A}}$ over diffusion randomness. Similarly,

$$\mathbf{Y}_{\mathcal{B}}(p) = \phi(\mathcal{B}(p, Z)) \sim \mathcal{Q}_{\mathcal{B}}(\cdot | p), \quad (10)$$

defines the corresponding prompt-conditioned semantic distribution of the compatible base model.

3.2 Dominant Hidden Concept Formulation

We characterize the dominant hidden concept from two complementary views: generation behavior and LoRA parameter space. A prompt revealing this concept should induce coherent LoRA outputs that are distinctive from those of the compatible base model and align strongly with the dominant input subspaces of the learned LoRA updates. We formalize these requirements as output coherence, LoRA-induced distinctiveness, and LoRA-subspace alignment.

Output coherence. A prompt revealing the dominant hidden concept should induce a coherent output distribution, with little variation across diffusion samples generated from the same prompt. Let $C(\cdot)$ denote a coherence functional over output distributions, with larger values indicating stronger coherence. We define the output coherence of prompt p as

$$\text{Coh}_{\mathcal{A}}(p) = C(\mathcal{Q}_{\mathcal{A}}(\cdot | p)). \quad (11)$$

LoRA-induced distinctiveness. Output coherence alone is insufficient, since a prompt may consistently produce a common concept that is already supported by the compatible base model. A prompt revealing the dominant hidden concept should therefore induce an output distribution distinct from that of the base model. Let $\mathcal{D}(\cdot, \cdot)$ denote a discrepancy functional between output distributions, with larger values indicating greater discrepancy. We define the LoRA-induced distinctiveness of prompt p as

$$\text{Dist}_{\mathcal{A}, \mathcal{B}}(p) = \mathcal{D}(\mathcal{Q}_{\mathcal{A}}(\cdot | p), \mathcal{Q}_{\mathcal{B}}(\cdot | p)). \quad (12)$$

LoRA-subspace alignment. Output coherence and LoRA-induced distinctiveness are still insufficient, since adaptation may also affect collateral concepts [22]. We assume that the primary training target is shared across concept-specific examples and is therefore expected to leave stronger evidence in the leading input directions than incidental content. A revealing prompt should consequently align strongly with these directions. Let $\mathcal{S}_{\mathcal{A}}$ denote the input subspaces spanned by the leading right singular directions of the LoRA updates, and let $\mathcal{H}(\cdot, \cdot)$ denote a prompt-subspace alignment functional, with larger values indicating stronger alignment. We define

$$\text{Align}_{\mathcal{A}}(p) = \mathcal{H}(p, \mathcal{S}_{\mathcal{A}}). \quad (13)$$

Dominant hidden concept. The three properties jointly characterize a prompt that reveals the dominant hidden concept. We define the dominant carrier prompt as

$$p^* \in \arg \max_{p \in \mathcal{P}} \left[\alpha \text{Coh}_{\mathcal{A}}(p) + \beta \text{Dist}_{\mathcal{A}, \mathcal{B}}(p) + \gamma \text{Align}_{\mathcal{A}}(p) \right], \quad (14)$$

where $\alpha, \beta, \gamma \geq 0$ balance the three components. The output distribution induced by p^* represents the discovered dominant hidden concept, while p^* serves as its carrier prompt.

3.3 Target-Concept Discrepancy

The formulation above defines the dominant hidden concept but does not quantify how faithfully a discovered carrier prompt reveals a given target concept. To provide such a measure, let c denote a target concept and P_c^{ref} its reference image distribution. For $X_c^{\text{ref}} \sim P_c^{\text{ref}}$, let

$$\mathbf{Y}_c^{\text{ref}} = \phi(X_c^{\text{ref}}) \sim \mathcal{Q}_c^{\text{ref}}, \quad (15)$$

where $\mathcal{Q}_c^{\text{ref}}$ is the corresponding distribution in the semantic representation space. We represent the target concept by the mean direction of its reference distribution:

$$\mu_c^{\text{ref}} = \mathbb{E}_{\mathbf{Y} \sim \mathcal{Q}_c^{\text{ref}}}[\mathbf{Y}], \quad \bar{\mu}_c^{\text{ref}} = \frac{\mu_c^{\text{ref}}}{\|\mu_c^{\text{ref}}\|_2}. \quad (16)$$

The target-concept discrepancy of the carrier prompt p^* is then defined as

$$\Delta_{\mathcal{A}}(p^*; c) = 1 - \mathbb{E}_{\mathbf{Y} \sim \mathcal{Q}_{\mathcal{A}}(\cdot | p^*)} \left[\mathbf{Y}^{\top} \bar{\mu}_c^{\text{ref}} \right]. \quad (17)$$

Since $\phi(\cdot)$ is ℓ_2 -normalized, Equation (17) measures the expected cosine distance between the LoRA-enabled outputs induced by p^* and the semantic direction of the target concept c . A smaller $\Delta_{\mathcal{A}}(p^*; c)$ therefore indicates that the discovered carrier prompt more faithfully reveals the target concept.

In evaluation, we estimate the reference direction from a finite set of reference images and average the cosine distance over generated outputs. This quantity is used only for evaluation and is unavailable during hidden-concept discovery.

3.4 Threat Model

We consider a model-sharing platform on which a creator uploads a LoRA adapter whose dominant hidden concept is not reliably disclosed [4, 19].

Creator capabilities. The creator controls the LoRA training process and may omit or obscure information about the concept encoded by the adapter, including its description, trigger words, and example images.

Platform capabilities. The platform moderator has white-box access to the uploaded adapter $\mathcal{A}$ and its compatible base model $\mathcal{B}$. The moderator may inspect the LoRA parameters and query both $\mathcal{B}_{\mathcal{A}}$ and $\mathcal{B}$, but has no access to the training

data, known trigger words, predefined concept queries, or reference images during discovery.

Moderator goal. The moderator aims to recover a carrier prompt p^* that satisfies the three desiderata formalized in Equation (14), using only information available from $\mathcal{A}$ and $\mathcal{B}$.

4 Methodology

4.1 Framework Overview

Equation (14) characterizes the carrier prompt through output coherence, LoRA-induced distinctiveness, and LoRA-subspace alignment. Directly optimizing these properties over the prompt space is impractical: the prompt space is combinatorially large, while evaluating the two output-side properties requires repeated diffusion rendering. We therefore develop SCOPE, a two-stage framework that separates render-free prompt search from output-side identification, as illustrated in Figure 1. SGES first uses an empirical alignment score to localize a high-alignment shortlist without diffusion rendering. PVI then renders only that shortlist, rejects candidates whose LoRA-enabled outputs are less coherent than the corresponding outputs from the compatible base model, and ranks the survivors using cross-prompt consensus together with an empirical distinctiveness score. The highest-ranked candidate is returned as the dominant carrier prompt p^* .

4.2 Subspace-Guided Evolutionary Search

SGES first addresses the LoRA-subspace alignment term in Equation (14). Unlike the two output-side properties, this alignment can be evaluated from the LoRA parameters and text-conditioned representations without diffusion rendering. SGES uses this signal to reduce the combinatorial prompt space to a compact set of high-alignment candidates for subsequent visual identification.

Text-conditioning sites. To obtain a render-free alignment signal, SGES focuses on LoRA sites whose inputs are determined by text conditioning and can be computed without executing the diffusion denoising trajectory. We consider LoRA sites in the text encoder and the key/value projections of text-facing cross-attention layers in the denoising backbone [32, 35]. For the U-Net-based models studied in this work, we group the retained sites into text-encoder te, down-path down, and up-path up groups. Let $\mathcal{J}_b$ denote the retained LoRA sites in group b , and define the groups present in adapter $\mathcal{A}$ as

$$\mathcal{G}_{\mathcal{A}} = \{b \in \{\text{te}, \text{down}, \text{up}\} : |\mathcal{J}_b| > 0\}. \quad (18)$$

denote the three site groups, and let $\mathcal{J}_b$ denote the retained LoRA sites in group $b \in \mathcal{G}_{\mathcal{A}}$. Architecture-specific site selection and grouping details are provided in Appendix A.1.

For a text input x and a retained site $s \in \mathcal{J}_b$, let

$$\mathbf{H}_s(x) = [\mathbf{h}_{s,1}(x), \dots, \mathbf{h}_{s,L(x)}(x)] \in \mathbb{R}^{d_s^{\text{in}} \times L(x)} \quad (19)$$

denote the text-conditioned input sequence at site s , where $L(x)$ is the tokenized sequence length and u indexes a token position. We denote the ℓ_2 -normalized representation at position u by $\hat{\mathbf{h}}_{s,u}(x)$.

Dominant input subspaces. For each retained LoRA site s , the learned weight update can be written as $\Delta \mathbf{W}_s = \mathbf{B}_s \mathbf{A}_s = \mathbf{U}_s \Sigma_s \mathbf{V}_s^\top$. For an input representation $\mathbf{h}$, the response $\Delta \mathbf{W}_s \mathbf{h}$ is determined by the projections of $\mathbf{h}$ onto the right singular vectors of $\Delta \mathbf{W}_s$. Accordingly, the right singular vectors associated with the largest singular values therefore define the dominant input subspace of the LoRA update. We retain the top- k directions $\{\mathbf{v}_{s,j}\}_{j=1}^k$ as the dominant input subspace at site s .

Site-position alignment. Given the dominant input directions at site s , we measure the alignment of each token position with these directions.

$$g_{s,u}(x) = \prod_{j=1}^k \left[\left(\mathbf{v}_{s,j}^\top \hat{\mathbf{h}}_{s,u}(x) \right)^2 \right]. \quad (20)$$

The product favors token positions that align consistently across the retained dominant directions, rather than exhibiting a strong projection onto only one direction.

Composed-word discovery. The tokenizer vocabulary consists of lexical units that may represent only fragments of a word [33, 38]. A candidate word may therefore span multiple tokenizer units. SGES performs composed-word discovery separately for each site group, combining promising units into candidate words before prompt-level search.

For a candidate word w composed of one to three lexical units, let $L(w)$ denote its tokenized sequence length. Because different LoRA sites may respond most strongly at different positions, we identify the position jointly supported by the sites in group b :

$$u_b^*(w) = \arg \max_{1 \leq u \leq L(w)} \left[\prod_{s \in \mathcal{J}_b} g_{s,u}(w) \right]^{1/|\mathcal{J}_b|}. \quad (21)$$

We then score the candidate by

$$S_b(w) = \frac{1}{|\mathcal{J}_b|} \sum_{s \in \mathcal{J}_b} g_{s,u_b^*(w)}(w). \quad (22)$$

This selection favors positions whose alignment is consistently supported across LoRA sites in the same group. SGES progressively extends high-scoring lexical units into two- and three-unit candidate words; detailed construction rules are provided in Appendix A.2.

Atomic candidate pools. SGES forms its prompt-search units from vocabulary words, the composed words discovered above, and common phrases. We refer to each such unit as an

SCOPE Open-ended LoRA hidden-concept discovery framework

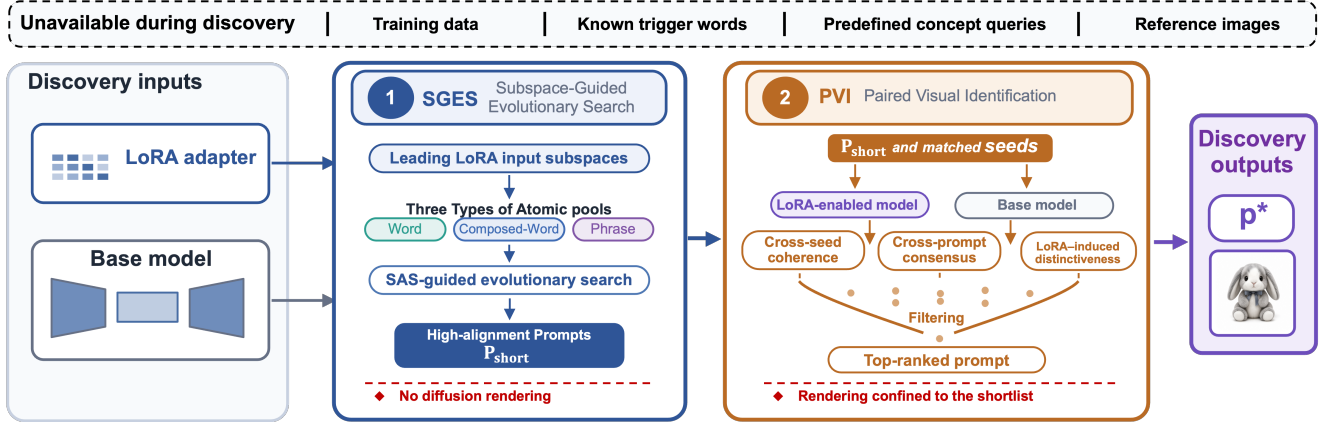


Figure 1: End-to-end SCOPE workflow. Training data, trigger words, predefined concept queries, and reference images are unavailable during discovery. Given only a LoRA adapter and its compatible base model, SGES extracts leading LoRA input subspaces, constructs Atomic pools, and performs SAS-guided evolutionary search to produce P_{short} without diffusion rendering. PVI renders the shortlisted prompts with the LoRA-enabled and base models under matched seeds, filters and ranks them using coherence, cross-prompt consensus, and LoRA-induced distinctiveness, and returns the top-ranked prompt p^* with its LoRA-enabled generated images.

Atomic. An Atomic may span multiple tokenizer positions but is treated as a single unit during prompt search.

For an Atomic a occurring in the text context x , let $\text{pos}_x(a)$ denote its token positions. Its response in site group b is defined as

$$R_b(a | x) = \frac{1}{|J_b|} \sum_{s \in J_b} \max_{u \in \text{pos}_x(a)} g_{s,u}(x). \quad (23)$$

Each LoRA site therefore contributes the position within a that aligns most strongly with its dominant input subspace.

To construct compact search pools, each Atomic is first evaluated in isolation using $R_b(a | a)$ and assigned to the site group in which it ranks highest. Each group retains its top-ranked Atomics, forming the candidate pool C_b . This reduces the lexical search space while preserving candidates that exhibit strong LoRA-subspace alignment. Pool construction details are provided in Appendix A.3.

Prompt-level alignment. LoRA-subspace alignment is context dependent, so Atomics that score highly in isolation may no longer receive high scores when combined into a prompt. SGES therefore re-evaluates each Atomic in the context of the complete prompt. Each present site group contributes three Atomics:

$$p = [a_b^i]_{b \in \mathcal{G}_{\mathcal{A}}, i=1,2,3}, \quad a_b^i \in C_b. \quad (24)$$

We instantiate the abstract alignment functional $\mathcal{H}(p, \mathcal{S}_{\mathcal{A}})$ in Equation (13) using the Subspace Alignment Score (SAS):

$$\widehat{\text{Align}}_{\mathcal{A}}(p) \triangleq \text{SAS}(p) = \frac{1}{3|\mathcal{G}_{\mathcal{A}}|} \sum_{b \in \mathcal{G}_{\mathcal{A}}} \sum_{i=1}^3 R_b(a_b^i | p). \quad (25)$$

SAS measures prompt-level LoRA-subspace alignment after contextual interaction among its Atomics.

Evolutionary prompt search. Although the Atomic pools substantially reduce the search space, exhaustive enumeration remains impractical because $\text{SAS}(p)$ depends on the complete prompt context. SGES therefore performs evolutionary search [27] using $\text{SAS}(p)$ as the fitness function. The resulting high-alignment prompts form the shortlist $\mathcal{P}_{\text{short}}$, which is passed to PVI for visual identification. The evolutionary operators and shortlist construction are detailed in Appendix A.4.

4.3 Paired Visual Identification

SGES reduces the full prompt space to candidates with strong LoRA-subspace alignment, but alignment alone does not guarantee that their generated outputs reveal a coherent LoRA-specific concept. PVI therefore evaluates the two output-side properties in Equation (14): output coherence and LoRA-induced distinctiveness. It first filters unstable LoRA responses, then uses cross-prompt consensus to suppress visual outliers, and finally combines consensus with LoRA-induced distinctiveness to identify the dominant carrier. Figure 2 mirrors these two stages.

Paired output samples. For each prompt $p \in \mathcal{P}_{\text{short}}$, PVI evaluates $\mathcal{B}_{\mathcal{A}}$ and $\mathcal{B}$ using the same R random seeds $\{z_r\}_{r=1}^R$. We denote the resulting realizations of $\mathbf{Y}_{\mathcal{A}}(p)$ and $\mathbf{Y}_{\mathcal{B}}(p)$ from Equations (9) and (10) by $\mathbf{y}_{\mathcal{A}}^r(p)$ and $\mathbf{y}_{\mathcal{B}}^r(p)$, respectively. Using matched seeds controls diffusion randomness when comparing the LoRA-enabled and base models [13].

Intra-prompt coherence filtering. We instantiate the co-

Paired Visual Identification (PVI)

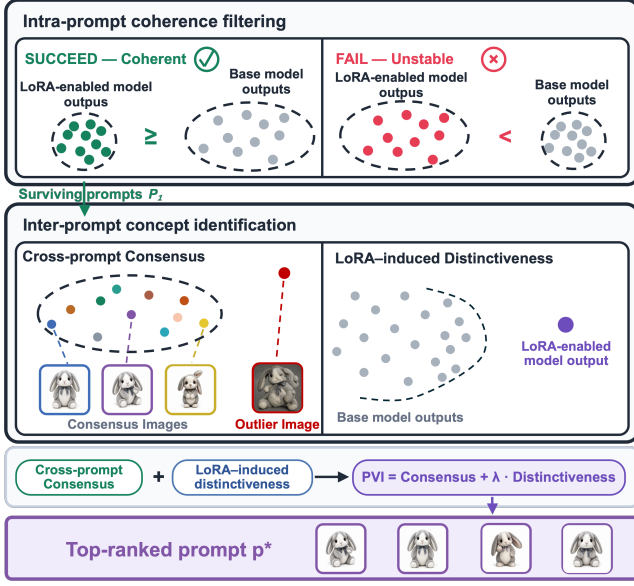


Figure 2: Two-stage PVI workflow. Intra-prompt coherence filtering first forms $\mathcal{P}_1$ by retaining prompts whose LoRA-enabled outputs are at least as coherent as the corresponding outputs from the compatible base model under matched seeds. Inter-prompt concept identification then ranks the survivors using two complementary signals: cross-prompt consensus down-ranks isolated candidates whose images are visually anomalous relative to the recurring cross-prompt pattern, while LoRA-induced distinctiveness favors prompts whose LoRA-enabled outputs are separated from the base-model outputs, providing evidence of LoRA-specific behavior. Their combined PVI score returns the top-ranked complete prompt p^* together with its LoRA-enabled generated images.

herence functional $\mathcal{C}(\cdot)$ in Equation (11) using the average pairwise cosine similarity across random generations. For $X \in \{\mathcal{A}, \mathcal{B}\}$, we define

$$\widehat{\text{Coh}}_X(p) \triangleq \frac{2}{R(R-1)} \sum_{1 \leq r < r' \leq R} (\mathbf{y}_X^r(p))^\top \mathbf{y}_X^{r'}(p). \quad (26)$$

A fixed coherence threshold is unsuitable because different concepts naturally exhibit different output variation. We therefore use the compatible base model under the same prompt as an adaptive reference. Since the adapter is trained to reinforce its encoded concept, a revealing prompt should yield LoRA-enabled outputs at least as coherent as the corresponding outputs from the compatible base model. We thus retain

$$\mathcal{P}_1 = \left\{ p \in \mathcal{P}_{\text{short}} \mid \widehat{\text{Coh}}_{\mathcal{A}}(p) \geq \widehat{\text{Coh}}_{\mathcal{B}}(p) \right\}. \quad (27)$$

Inter-prompt concept identification. After intra-prompt filtering, let $\mathcal{P}_1 = \{p_1, \dots, p_M\}$ denote the surviving candidates.

Although their LoRA-enabled outputs are stable, some candidates may still be visual outliers. Such outliers can appear highly distinctive from the base model without representing the dominant concept. PVI therefore combines cross-prompt consensus with LoRA-induced distinctiveness for concept identification.

For each candidate p_i , we define its cross-prompt consensus as

$$\text{Con}_{\mathcal{A}}(p_i) = \frac{1}{M-1} \sum_{j \neq i} \frac{1}{R} \sum_{r=1}^R (\mathbf{y}_{\mathcal{A}}^r(p_i))^\top \mathbf{y}_{\mathcal{A}}^r(p_j). \quad (28)$$

A larger $\text{Con}_{\mathcal{A}}(p_i)$ indicates stronger agreement with the other surviving candidates and thus a lower likelihood of being a visual outlier. Outputs are compared under matched random seeds to reduce variation from diffusion randomness.

We next instantiate the LoRA-induced distinctiveness in Equation (12) using the average cosine dissimilarity between the LoRA-enabled outputs of p_i and the base-model outputs across the surviving prompts:

$$\widehat{\text{Dist}}_{\mathcal{A}, \mathcal{B}}(p_i) \triangleq \frac{1}{M} \sum_{j=1}^M \frac{1}{R} \sum_{r=1}^R \left[1 - (\mathbf{y}_{\mathcal{A}}^r(p_i))^\top \mathbf{y}_{\mathcal{B}}^r(p_j) \right]. \quad (29)$$

A larger $\widehat{\text{Dist}}_{\mathcal{A}, \mathcal{B}}(p_i)$ indicates stronger separation from the base-model outputs.

PVI combines cross-prompt consensus with the empirical distinctiveness score above:

$$\text{PVI}(p_i) = \text{Con}_{\mathcal{A}}(p_i) + \lambda \widehat{\text{Dist}}_{\mathcal{A}, \mathcal{B}}(p_i), \quad (30)$$

where $\lambda \geq 0$ balances the two terms. The highest-scoring candidate is returned as the dominant carrier prompt p^* .

5 Evaluation

5.1 Evaluation Setup

Benchmark and models. We evaluate SCOPE on LoRA adapters from the SD1.5 and SDXL model families [32, 35, 43, 44]. The benchmark spans real identities, synthetic identities, anime characters, and artistic styles, and combines public adapters [4, 19] with identity adapters trained under controlled settings. Each adapter is paired with a compatible base model from the same model family. Appendix B.1 gives the benchmark composition; Appendices B.2 and B.3 give the local training and data-construction procedures. We keep these cohort details in the appendix and state below whenever a result uses a restricted subset.

Baselines. Hidden-concept discovery requires identifying an unknown concept without a predefined concept, and few existing methods directly address this setting. To enable broad comparison, we evaluate seven representative methods from closely related tasks. These methods can be broadly grouped into two categories:

- *Concept-agnostic difference discovery.* DriftScope [22], CompCon [8], VisDiff [10], and DPO-Diff-EPO [50] share the goal of exposing differences between model behaviors without assuming a predefined concept. We adapt them to compare the LoRA-enabled model with its compatible base model and interpret the discovered LoRA-specific differences as candidate hidden concepts.
- *Concept-specified verification.* PEZ [52] recovers prompts from example images of a specified concept, which we use to query the LoRA-enabled model. PAIA [55] instead directly verifies whether a specified concept is encoded by comparing the denoising behavior of the LoRA-enabled and base models. All three methods require the concept to be specified beforehand, so their coverage is limited to the candidate concepts examined.

Detailed baseline adaptations are provided in Appendix B.4. **Evaluation metric.** Equation (17) defines the discrepancy between the outputs induced by a discovered carrier prompt and the semantic direction of a target concept. In evaluation, let $\mathcal{Y}_{p^*} = \{\mathbf{y}_i\}_{i=1}^n$ and $\mathcal{R}_c = \{\mathbf{r}_j\}_{j=1}^m$ denote finite samples from $Q_{\mathcal{A}}(\cdot | p^*)$ and Q_c^{ref} , respectively. We estimate the reference concept direction by

$$\hat{\mu}_c^{\text{ref}} = \frac{1}{m} \sum_{j=1}^m \mathbf{r}_j, \quad \bar{\mu}_c^{\text{ref}} = \frac{\hat{\mu}_c^{\text{ref}}}{\|\hat{\mu}_c^{\text{ref}}\|_2}. \quad (31)$$

The empirical target-concept discrepancy is then

$$\hat{\Delta}_{\mathcal{A}}(p^*; c) \triangleq 1 - \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i^\top \bar{\mu}_c^{\text{ref}}. \quad (32)$$

Since the semantic representations are ℓ_2 normalized, $\hat{\Delta}_{\mathcal{A}}(p^*; c)$ is the average cosine distance between the generated outputs and the estimated semantic direction of concept c . A smaller discrepancy indicates that the discovered carrier more faithfully reveals the target concept.

Implementation details. SGES retains the top $k = 2$ right singular directions at each selected LoRA site. Each present site group retains 50 Atomic candidates, including 30 token candidates and 20 phrase candidates. Evolutionary search uses a population of 48 for 100 generations and is repeated over three independent runs. PVI uses $R = 5$ paired generation seeds and sets $\lambda = 0.5$. We use `openai/clip-vit-large-patch14` [29, 33] for PVI and final evaluation. For each evaluated prompt, we generate $n = 10$ images and separately generate $m = 10$ held-out references using disjoint random seeds; references are never available during discovery. All image generation follows the compatible configuration of each benchmark case. Experiments run on NVIDIA A40 GPUs. The precise text-conditioning sites, composed-word procedure, Atomic-pool construction, and evolutionary shortlist are documented in Appendices A.1–A.4.

Table 1: Hidden-concept discovery on the common faithful SD1.5 cohort. Concept Input indicates access to target examples. Entries report mean target-concept discrepancy within each category; Mean is adapter-weighted across adapters. Lower is better.

Method	Concept	Synthetic			
	Input	Real ID ↓	ID ↓	Anime ↓	Mean ↓
DriftScope	×	0.451	0.437	0.416	0.435
CompCon	×	0.543	0.462	0.404	0.475
VisDiff	×	0.396	0.333	0.393	0.383
DPO-Diff-EPO	×	0.552	0.449	0.490	0.509
PEZ	✓	<u>0.171</u>	<u>0.196</u>	<u>0.190</u>	<u>0.183</u>
Ours	×	0.078	0.178	0.123	0.114

5.2 Main Results

5.2.1 Hidden-Concept Discovery

Table 1 compares the semantic fidelity of the returned prompts using the target-concept discrepancy in Equation (17). DriftScope, CompCon, VisDiff, DPO-Diff-EPO, and SCOPE receive no predefined concept during search. PEZ is instead supplied with target-concept images, providing a reference-assisted comparison.

Results. SCOPE achieves the lowest discrepancy for every category, with an adapter-weighted mean of 0.114. VisDiff is the strongest comparison method under the same no-concept-input setting, with a mean of 0.383; SCOPE reduces this discrepancy by 70.2%. PEZ benefits substantially from its target images and reaches 0.183, yet SCOPE further reduces its discrepancy by 37.7% without receiving a concept name, trigger, or target example.

The category-level results follow the same ordering. For real identities, SCOPE reaches 0.078, compared with 0.396 for VisDiff and 0.171 for PEZ. For synthetic identities, it reaches 0.178, improving on 0.333 for VisDiff and 0.196 for PEZ. For anime characters, it reaches 0.123, compared with 0.393 and 0.190, respectively. Thus, the overall gain is supported by all three categories rather than being driven by one concept type.

Analysis. The separation from the concept-agnostic baselines highlights the difference between exposing an arbitrary LoRA–base change and identifying the adapter’s dominant hidden concept. A LoRA can alter several concepts or visual attributes, so a prompt that maximizes a behavioral difference need not reveal the encoded target concept. SCOPE adds two complementary sources of selectivity: SGES prioritizes prompts supported by the leading input subspaces of the LoRA updates, and PVI retains coherent responses that recur across high-alignment prompts and remain distinctive from the compatible base model. The component results in Section 5.3.1 directly isolate these contributions.

PEZ provides an informative second comparison. Its

Table 2: Concept-specificity verification with one target and one fixed non-target query per adapter. PEZ and PAIA receive the queried concept, whereas SCOPE uses its fixed discovered prompt. Higher is better.

Method	Accuracy $\uparrow$	Precision $\uparrow$	Recall $\uparrow$	F1 $\uparrow$
PEZ	0.941	0.959	0.922	0.940
PAIA	0.951	0.911	1.000	0.953
Ours	0.980	0.980	0.980	0.980

improvement over the concept-agnostic baselines confirms that target images offer strong guidance for prompt recovery. SCOPE nevertheless achieves a lower discrepancy in every category without this guidance. This result is consistent with the paper’s central premise: the adapter parameters themselves contain useful evidence for localizing the dominant concept, and output-side identification can turn that evidence into a prompt whose generations match the target. The comparison therefore tests the deployment setting directly: whether an unknown target can be localized from a submitted adapter, rather than whether a supplied target can be reproduced.

5.2.2 Concept-Specificity Verification

The discrepancy experiment above measures how closely a returned prompt reveals the target. We next test the complementary property of concept specificity: whether the same evidence distinguishes the target from a non-target concept instead of merely expressing a broad category or incidental attribute. For each adapter, we form one positive query using its ground-truth concept and one negative query using a deterministically selected non-target concept. This produces a balanced evaluation with identical queries for all methods.

PEZ and PAIA receive the queried concept before making each decision. In contrast, SCOPE first discovers p^* without a concept query and then uses that same fixed prompt for both its target and non-target decision. For PEZ and SCOPE, we apply the pre-specified threshold $\tau = 0.3$ to the target-concept discrepancy:

$$\hat{y}(\mathcal{A}_i, c) = \mathbb{1} \left[\hat{\Delta}_{\mathcal{A}_i}(p; c) \leq \tau \right], \quad \tau = 0.3. \quad (33)$$

Here $p = p^*$ for SCOPE and is the prompt recovered from the supplied query images for PEZ. PAIA uses its native binary decision rule.

Results. Table 2 shows that SCOPE achieves the highest accuracy, precision, and F1, all at 0.980, while also reaching 0.980 recall. PAIA attains perfect recall but has lower precision of 0.911, indicating that its operating point favors positive decisions. PEZ has higher precision of 0.959 than recall of 0.922, showing the opposite imbalance. In comparison, SCOPE obtains equal precision and recall while improving

overall accuracy and F1. The advantage therefore comes from treating target and non-target decisions more evenly rather than optimizing only one side of the binary task.

Analysis. These results complement the continuous fidelity measurement in Table 1. A low discovery discrepancy shows that the returned prompt reveals the intended concept, while high specificity additionally requires rejecting a controlled non-target. The 0.980 precision and recall show that the fixed p^* supports both properties. In particular, the result is obtained without query-conditioned search: SCOPE discovers the prompt once and then evaluates it against both concepts.

PEZ and PAIA begin from the concept being tested, whereas SCOPE must first determine what concept the adapter predominantly encodes. Its higher accuracy and F1 therefore show that open-ended discovery can produce evidence useful for a subsequent concept-specific decision. Together, Tables 1 and 2 establish two distinct properties of the returned prompt: semantic fidelity to the hidden target and specificity against a non-target concept.

5.2.3 Composition of the Top-Ranked Prompt

Figure 3 shows how the content of the top-ranked prompt relates to the training trigger and encoded concept. For Chun-Li, the complete `chunli` string occurs inside several Composed-word Atomics; for Brad Pitt, `brad` and `pitt` recur inside different Composed-word Atomics. These examples demonstrate the value of BPE-aware composition for names represented by multiple lexical units. For Elise Rowan, both trigger words are recovered as separate Word Atomics. The Watercolor example follows a complementary semantic route, combining `watercolour` with target-related descriptions.

Because SGES selects Atomics by alignment with the dominant LoRA input subspaces, literal fragments and semantic descriptions can jointly activate the encoded concept. Exact trigger recovery is not the discovery objective: a carrier prompt succeeds when its LoRA-enabled outputs reveal the encoded concept. Semantically related or partially overlapping Atomics can therefore be sufficient even when p^* does not reconstruct the training trigger.

5.2.4 Visual Discovery Results

Figure 4 makes the quantitative result visually concrete. For both Chun-Li and Julian Frost, generations from SCOPE’s top-ranked prompts preserve the defining identity visible in the references. PEZ is the closest comparison method and is additionally given target-concept images; in the anime example, however, it recovers the anime domain rather than the same character. In the examples shown, the remaining methods return different identities, characters, or incidental visual patterns. In contrast, SCOPE reveals both hidden identities without any target-concept information, showing that the parameter-guided search recovers the dominant LoRA

Atomic type: Word Composed word Phrase		Relation to trigger: Literal trigger overlap Target-related Unrelated/Noise	
Category	Trigger token	Top-ranked prompt	
Anime			
Chun-Li	ChunLi	chunligirl breasts chunchunli girl from girl portrait chun chunli	
Real ID			
Brad Pitt	brad pitt	picture se bradldepitt braddepitt and brad deborhttp bradhttp dark and dailbtn	
Synthetic ID			
Elise Rowan	elise rowan	elise rowan elsanne else	
Style			
Watercolor	watercolor painting	bldg tree forest with snowball watercolour magical forest forest landscape forest by	

Figure 3: Composition of representative top-ranked prompts. The first column reports each LoRA category and concept, the second gives its training trigger revealed only after discovery, and the third decomposes p^* returned by SCOPE. Solid blue, dashed purple, and dotted ochre borders denote Word, Composed-word, and Phrase Atomics. In p^* , red text marks literal trigger overlap, italic green text marks target-related descriptions, and black text marks unrelated/noise. All annotations are added post hoc and are never used by discovery.

concept rather than only a conspicuous output difference.

5.2.5 Discovery Across Backbones and Concept Types

Figure 5(b) shows that the same discovery procedure produces concept-aligned prompts on both SD1.5 and SDXL, with all concept-type means below 0.3. For real identities, synthetic identities, and anime characters, discrepancies are 0.078, 0.178, and 0.123 on SD1.5 and 0.111, 0.212, and 0.173 on SDXL. Style discovery reaches 0.264 on SD1.5 and 0.249 on SDXL, extending the method to concepts defined by shared visual attributes rather than one identity.

Figure 5(a) makes this result visible. The oil-painting prompts reproduce painterly texture and color treatment across different subjects, while the stained-glass prompts reproduce segmented geometry, dark leading, and luminous color across different scenes. The returned prompts therefore recover the shared high-level style instead of copying one reference subject.

5.3 Ablation Study

We isolate the contribution of the two pipeline stages, the LoRA input directions used by SGES, contextual prompt search, and site-specific Atomic pools. Appendix C.1 additionally evaluates the retained subspace dimension.

5.3.1 Impact of the Core Components

CW denotes composed-word discovery; Coh., Cons., and Dist. denote intra-prompt coherence filtering, cross-prompt consensus, and LoRA-induced distinctiveness. Composed-word discovery first enriches the lexical search space: removing it raises the mean from 0.114 to 0.122, with the largest change for real identities (0.078 to 0.092). This agrees with

Table 3: Contributions of SGES, PVI, composed-word discovery, and the three PVI operations. Each row removes one component and reruns discovery; lower target-concept discrepancy is better.

Variant	Real ID ↓	Synthetic ID ↓	Anime ↓	Mean ↓
w/o SGES	0.472	0.313	0.344	0.393
w/o PVI	0.096	0.192	0.138	0.130
w/o CW	0.092	0.180	0.126	0.122
w/o Coh.	0.087	0.187	0.131	0.123
w/o Cons.	0.164	0.238	0.242	0.208
w/o Dist.	0.083	0.179	0.128	0.119
Full	0.078	0.178	0.123	0.114

Figure 3, where multi-unit names are recovered through composed Atomics.

SGES then converts this richer vocabulary into parameter-supported prompt candidates. Removing SGES and sampling from the raw lexical source raises the mean to 0.393, the largest change in Table 3. PVI acts on the resulting SGES shortlist; removing it raises the mean to 0.130, showing that output-side identification adds a consistent gain after parameter-guided search.

The PVI ablations follow its processing order. Intra-prompt coherence filtering removes unstable candidates and improves the mean from 0.123 to 0.114. Cross-prompt consensus provides the largest identification gain: without it, the mean rises to 0.208 and anime discrepancy rises from 0.123 to 0.242, confirming the value of down-ranking visual outliers that do not recur across other high-alignment prompts. LoRA-induced distinctiveness then favors outputs introduced by the adapter, improving the mean from 0.119 to 0.114. Together, the components form a connected progression from BPE-aware candidates, to parameter-guided localization, to visual



Figure 4: Visual comparison of hidden-concept discovery for an anime character and a synthetic identity. Reference images are generated by the LoRA-enabled model under verified trigger prompts; the remaining columns show LoRA-enabled generations under each method’s returned prompt. PEZ receives target-concept images, whereas SCOPE and the other displayed methods receive no target-concept input.

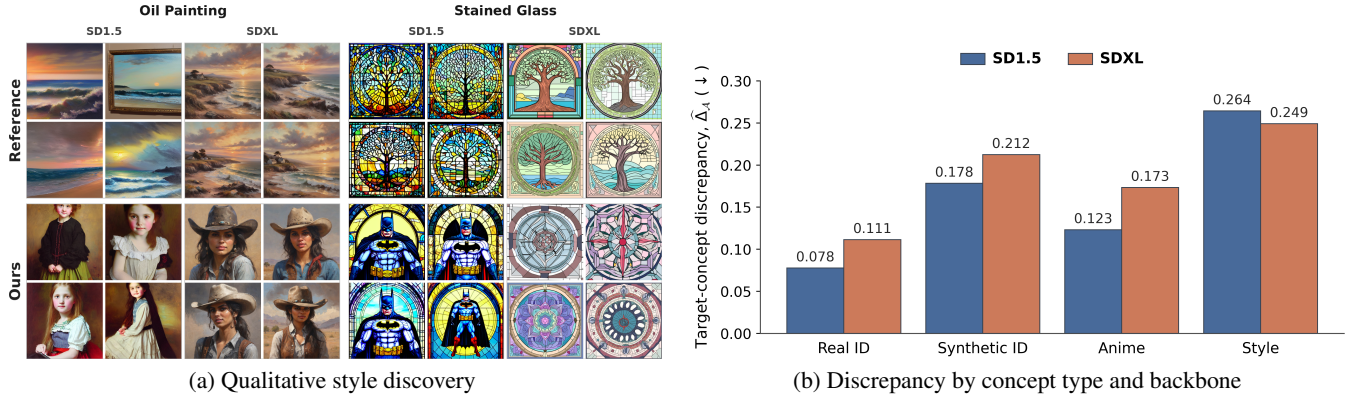


Figure 5: Discovery across backbones and concept types. (a) Trigger-prompt generations (top) and generations from SCOPE’s top-ranked prompts (bottom) for two styles on SD1.5 and SDXL. (b) Mean target-concept discrepancy by concept type and backbone; lower is better.

Table 4: Effect of LoRA input-direction selection at $k = 2$. Random uses a dimension-matched orthonormal subspace; lower target-concept discrepancy is better.

Direction	Real ID ↓	Synthetic ID ↓	Anime ↓	Mean ↓
Random	0.396	0.258	0.301	0.334
Trailing	0.485	0.343	0.327	0.398
Leading	0.078	0.178	0.123	0.114

identification.

5.3.2 Impact of the LoRA Subspace Selection

This experiment tests the parameter-space signal at the center of SGES. We fix $k = 2$ and replace the leading right-singular directions with either the final two directions in the effective

update subspace or a dimension-matched random orthonormal subspace.

Table 4 gives the same strict ordering in every category. Leading directions achieve a mean discrepancy of 0.114, compared with 0.334 for random directions and 0.398 for trailing directions. Because the random control has the same dimension, low dimensionality alone does not explain the gain. Singular-value ordering concentrates the LoRA evidence that SGES uses to localize hidden-concept prompts. The dimension sensitivity in Appendix C.1 further gives means of 0.128, 0.114, and 0.127 for $k = 1, 2, 4$, respectively; $k = 2$ is best, and the tested range varies by at most 0.014.

5.3.3 Impact of the Search Strategy

This experiment holds the group-specific Atomic pools, short-list budget, and PVI stage fixed and changes only how com-

Table 5: Prompt-search strategies over identical group-specific Atomic pools with fixed shortlist and PVI budgets. Lower target-concept discrepancy is better.

Strategy	Real ID ↓	Synthetic ID ↓	Anime ↓	Mean ↓
Greedy	0.148	0.217	0.180	0.173
Random	0.107	0.191	0.138	0.135
Evolutionary	0.078	0.178	0.123	0.114

Table 6: Contribution of text-encoder (TE), U-Net up-path (UP), and U-Net down-path (DOWN) site groups. One Bucket merges candidates from all retained sites; lower target-concept discrepancy is better.

Variant	Real ID ↓	Synthetic ID ↓	Anime ↓	Mean ↓
One Bucket	0.097	0.183	0.156	0.136
w/o TE	0.091	0.180	0.150	0.131
w/o UP	0.136	0.250	0.203	0.183
w/o DOWN	0.112	0.271	0.133	0.151
Full	0.078	0.178	0.123	0.114

plete prompts are selected. Greedy combines the strongest standalone Atomics, Random samples from the already constructed Atomic pools, and Evolutionary re-encodes each complete prompt and optimizes its contextual SAS. Thus, Random here differs from *w/o SGES* in Table 3: it retains SGES’s parameter-enriched Atomic pools rather than sampling from the raw lexical source.

Table 5 shows the same ordering in every category. Random reaches 0.135, demonstrating that the group-specific Atomic pools already contain useful signal. Greedy reaches 0.173 because Atomics that align well alone need not remain optimal after contextual encoding. Evolutionary search reaches 0.114 by optimizing the complete prompt in context, improving on both fixed alternatives.

5.3.4 Impact of the Site Grouping

Table 6 separates access to LoRA sites from the organization of their evidence. One Bucket retains all selected sites but merges their candidates, increasing mean discrepancy from 0.114 to 0.136. Separate pools therefore add value even when no site is removed: they preserve complementary evidence and prevent one pathway from dominating prompt construction.

The removal results show how those contributions combine. Excluding UP has the largest overall effect, increasing the mean to 0.183. Excluding DOWN is especially costly for synthetic identities, increasing their discrepancy from 0.178 to 0.271, while excluding TE also degrades every category. These distinct category profiles explain why the full method constructs prompts from all available site groups.

Table 7: Discovery under matched human-readable and opaque-trigger training. Each pair shares its data and optimization settings and differs only in the trigger; Δ_{trig} is opaque minus human-readable discrepancy. Lower is better.

Model	Concept	Human-readable ↓	Opaque ↓	Δ_{trig} ↓
SD1.5	Real ID	0.078	0.224	+0.147
	Synthetic ID	0.178	0.292	+0.114
	Overall	0.112	0.247	+0.136
SDXL	Real ID	0.111	0.204	+0.093
	Synthetic ID	0.212	0.358	+0.145
	Overall	0.146	0.257	+0.111
Combined		0.129	0.252	+0.123

5.4 Robustness to Concealed Triggers and Adaptive Training

We evaluate two increasingly targeted challenges: an opaque training trigger that removes semantic disclosure, and adaptive training that disperses the dominant LoRA input subspaces used by SGES.

5.4.1 Discovery Under Opaque Triggers

Each controlled identity is represented by a matched human-readable and opaque-trigger adapter on both backbones. A pair shares training images and optimization settings and differs only in its caption trigger. Neither trigger is available during discovery. For each pair, we define

$$\Delta_{\text{trig}} = \hat{\Delta}_{\mathcal{A}_{\text{opaque}}}(p_{\text{opaque}}^*; c) - \hat{\Delta}_{\mathcal{A}_{\text{readable}}}(p_{\text{readable}}^*; c). \quad (34)$$

Table 7 shows that replacing readable names with opaque strings increases the combined mean discrepancy from 0.129 to 0.252. Nevertheless, the opaque-trigger means remain 0.247 on SD1.5 and 0.257 on SDXL, and the combined mean remains below the evaluation threshold of 0.3. Thus, SCOPE continues to expose substantial concept evidence even when the training trigger provides no human-readable semantics. Because neither trigger is supplied during discovery, this robustness comes from evidence encoded in the adapter rather than direct access to the trigger text.

5.4.2 Adaptive Training: Subspace Dispersion

We next evaluate a creator who knows that SGES uses the dominant LoRA input subspaces and adds a fixed spectral-dispersion regularizer during training. The creator does not know the retained dimension, inspected sites, or other discovery hyperparameters. The attack counts as a successful evasion only when the creator’s fixed trigger prompt still reveals the target but the top-ranked prompt returned by SCOPE does

Table 8: Adaptive subspace-dispersion attack. Prompt columns report mean target-concept discrepancy under attacked adapters. Conditional evasion is measured only among adapters whose fixed trigger prompt preserves target generation.

Attack	Trigger Prompt ↓	Top-ranked Prompt ↓	Utility Preserved ↑	Conditional Evasion ↓
Dispersion	0.153	0.162	86.2%	4.0%

not. Appendix D gives the threat model, objective, schedule, and matched controls.

For an attacked adapter, p_{trig} denotes the fixed generation prompt containing its human-readable training trigger; it is available to the creator but unavailable during discovery. SCOPE instead returns p^* . *Utility Preserved* is the fraction satisfying $\hat{\Delta}_{\mathcal{A}}(p_{\text{trig}}; c) \leq 0.3$. *Conditional Evasion* is the fraction of these utility-preserving adapters for which $\hat{\Delta}_{\mathcal{A}}(p^*; c) > 0.3$.

Attack effectiveness. The mean spectral-concentration statistic in Equation (44) falls from 0.127 to 0.013, an 89.8% reduction. The attack therefore substantially flattens the LoRA spectrum and directly weakens the structure SGES is designed to use.

Utility-preserving evasion. The attacked adapters retain target-generation utility in 86.2% of cases. Their mean trigger-prompt and top-ranked-prompt discrepancies are 0.153 and 0.162, respectively. Although dispersion raises the top-ranked-prompt discrepancy from 0.112 on the matched clean cohort to 0.162, conditional evasion among utility-preserving adapters is only 4.0%. Under this fixed attack, even an 89.8% reduction in spectral concentration rarely conceals the target while preserving the creator’s ability to generate it.

6 Discussion

Why parameter-guided discovery works. The experiments show that LoRA parameters provide useful evidence for open-ended concept discovery even when the concept name, training trigger, and reference images are unavailable. SGES uses the dominant LoRA input subspaces to concentrate search on parameter-supported prompts. PVI then applies intra-prompt coherence filtering, cross-prompt consensus, and LoRA-induced distinctiveness to select the prompt whose outputs most consistently reveal the adapter-specific concept. The ablations connect these roles: removing SGES produces the largest overall degradation, while cross-prompt consensus is the most important operation within PVI.

Evidence for platform review. SCOPE returns a top-ranked carrier prompt together with its generated images. A reviewer can therefore inspect both the lexical evidence and the visual concept it elicits, even when the prompt does not reproduce the original trigger. This output is designed to support mod-

eration and downstream investigation rather than make an enforcement decision by itself. Failure to discover a carrier prompt is not evidence that an adapter is benign or contains no undisclosed concept.

7 Limitations

We evaluate separately loaded, U-Net-based LoRAs for SD1.5 and SDXL under white-box access to the adapter and its compatible base model. The current implementation searches short English-language, ASCII-only Atomics at text-conditioned LoRA sites. The results therefore do not establish performance for DiT-based generators, merged or quantized adapters, black-box access, or multilingual and nonlexical carriers.

The formulation seeks one dominant hidden concept rather than an exhaustive inventory. Adapters containing several comparably strong concepts, or concepts deliberately distributed outside the inspected subspaces, may require repeated or broader discovery. Evaluation uses CLIP representations and one mean reference direction, verification uses one controlled non-target per adapter, and the adaptive study tests one fixed subspace-dispersion schedule. Broader safety concepts, attacks, independent semantic encoders, human evaluation, and runtime measurement remain important before deployment.

8 Conclusion

We formulate open-ended hidden-concept discovery for LoRA adapters and develop SCOPE, which combines render-free parameter-guided search with output-based visual identification. On the faithful SD1.5 cohort, SCOPE reaches a mean target-concept discrepancy of 0.114, reducing discrepancy by 70.2% relative to the strongest baseline with the same no-concept-input setting and by 37.7% relative to reference-assisted PEZ. Its discovered prompts further attain 0.980 precision, recall, and F1 in controlled target-specificity evaluation. Under adaptive subspace dispersion, conditional evasion remains 4.0% among adapters that preserve target-generation utility. More broadly, the evaluation separates open-ended discovery from concept-specified verification. SCOPE does not assume what a reviewer should test; it first extracts parameter-supported candidates and then confirms them through generated evidence. This matches the model-sharing setting, where the concealed concept is unknown before inspection, and makes the returned prompt–image pair directly useful for review rather than only reporting a model-level score. These results demonstrate that LoRA parameter structure can guide discovery of previously unknown concepts and produce concrete prompts and images for subsequent human review.

References

- [1] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [2] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pages 5253–5270, 2023.
- [3] 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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829. IEEE, 2023.
- [4] Civitai. Models. Civitai model-sharing platform. <https://civitai.com/models>. Accessed October 12, 2026.
- [5] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10850–10869, 2023.
- [6] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.
- [7] Lisa Dunlap, Joseph E. Gonzalez, Trevor Darrell, Fabian Caba Heilbron, Josef Sivic, and Bryan Russell. CompCon: Official implementation. GitHub repository. <https://github.com/adobe-research/CompCon>. Accessed October 12, 2026.
- [8] Lisa Dunlap, Joseph E Gonzalez, Trevor Darrell, Fabian Caba Heilbron, Josef Sivic, and Bryan Russell. Discovering divergent representations between text-to-image models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17516–17525. IEEE, 2025.
- [9] Lisa Dunlap, Yuhui Zhang, Xiaohan Wang, Ruiqi Zhong, Trevor Darrell, Jacob Steinhardt, Joseph E. Gonzalez, and Serena Yeung-Levy. VisDiff: Official implementation. GitHub repository. <https://github.com/Understanding-Visual-Datasets/VisDiff>. Accessed October 12, 2026.
- [10] Lisa Dunlap, Yuhui Zhang, Xiaohan Wang, Ruiqi Zhong, Trevor Darrell, Jacob Steinhardt, Joseph E Gonzalez, and Serena Yeung-Levy. Describing differences in image sets with natural language. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24199–24208. IEEE, 2024.
- [11] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (EC) no 300/2008, (EU) no 167/2013, (EU) no 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act). Official Journal of the European Union, L 2024/1689. <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>, July 2024. Article 50; accessed October 12, 2026.
- [12] Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *European Conference on Computer Vision*, pages 172–188. Springer, 2024.
- [13] Paul Glasserman and David D Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992.
- [14] Gustavosta. Gustavosta/Stable-Diffusion-Prompts. Hugging Face dataset card. <https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts>. Accessed October 12, 2026.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [17] 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 *International conference on machine learning*, pages 2790–2799. PMLR, 2019.
- [18] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [19] Hugging Face. The model hub. Hugging Face Hub Documentation. <https://huggingface.co/docs/hub/main/models-the-hub>. Accessed October 12, 2026.
- [20] Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. Datadream: Few-shot guided dataset generation. In *European Conference on Computer Vision*, pages 252–268. Springer, 2024.
- [21] Héctor Laria, Yiping Han, Julian D. Santamaria, Kai Wang, Bogdan Raducanu, Joost van de Weijer, and Alexandra Gomez-Villa. DriftScope: Official implementation. GitHub repository. <https://github.com/hyping11/DriftScope>. Accessed October 12, 2026.
- [22] Héctor Laria, Yiping Han, Julian D. Santamaria, Kai Wang, Bogdan Raducanu, Joost van de Weijer, and Alexandra Gomez-Villa. Driftscope: Measuring the hidden effects of diffusion model adaptation. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari, editors, *Computer Vision – ECCV 2026*, pages 58–78, Cham, 2026. Springer Nature Switzerland.
- [23] 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*, pages 3045–3059, 2021.
- [24] 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 (volume 1: Long papers)*, pages 4582–4597, 2021.
- [25] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in neural information processing systems*, 35:1950–1965, 2022.
- [26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [27] Brad L Miller and David E Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212, 1995.
- [28] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 17–23 Jul 2022.

- [29] OpenAI. openai/clip-vit-large-patch14 model card. Hugging Face model card. <https://huggingface.co/openai/clip-vit-large-patch14>. Accessed October 12, 2026.
- [30] OpenAI. Introducing GPT-5.5. <https://openai.com/index/introducing-gpt-5-5/>, April 2026. Published April 23, 2026; updated April 24, 2026; accessed October 12, 2026.
- [31] Alain Pétrowski. A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE international conference on evolutionary computation*, pages 798–803. IEEE, 1996.
- [32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pages 1862–1874, 2024.
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.
- [36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [37] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22522–22531. IEEE, 2023.
- [38] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725, 2016.
- [39] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by Text-to-Image models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2187–2204, Anaheim, CA, August 2023. USENIX Association.
- [40] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-LoRA. *Transactions on Machine Learning Research*, 2024.
- [41] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.
- [42] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6048–6058. IEEE, 2023.
- [43] Stability AI. SD-XL 1.0-base Model Card. Hugging Face model card. <https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0>. Accessed October 12, 2026.
- [44] Stable Diffusion v1-5. Stable Diffusion v1-5 Model Card. Hugging Face model card. <https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5>. Accessed October 12, 2026.
- [45] United States Congress. Tools to Address Known Exploitation by Immobilizing Technological Deepfakes on Websites and Networks Act (TAKE IT DOWN Act). U.S. Government Publishing Office, Public Law 119–12, 139 Stat. 55. <https://www.govinfo.gov/app/details/PLAW-119publ12>, May 2025. Approved May 19, 2025; accessed October 12, 2026.
- [46] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2116–2127. IEEE, 2023.
- [47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [48] Laura Wagner and Eva Cetinić. Perpetuating misogyny with generative ai: How model personalization normalizes gendered harm. *Big Data & Society*, 13(3):20539517261467361, 2026.
- [49] Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. DPO-Diff: Official implementation. GitHub repository. <https://github.com/ruocwang/dpo-diffusion>. Accessed October 12, 2026.
- [50] Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. On discrete prompt optimization for diffusion models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 50992–51011. PMLR, 21–27 Jul 2024.
- [51] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard Prompts Made Easy: Official implementation. GitHub repository. <https://github.com/YuxinWenRick/hard-prompts-made-easy>. Accessed October 12, 2026.
- [52] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36:51008–51025, 2023.
- [53] Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. DiffFit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4207–4216. IEEE, 2023.
- [54] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *2024 IEEE symposium on security and privacy (SP)*, pages 897–912. IEEE, 2024.
- [55] Xiaoyong Yuan, Xiaolong Ma, Linke Guo, and Lan Zhang. What lurks within? concept auditing for shared diffusion models at scale. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 4154–4168, 2025.

A Additional Details of SGES

A.1 Text-Conditioning Site Selection

SGES retains only LoRA sites whose inputs can be determined from text conditioning without executing the diffusion denoising trajectory. This allows LoRA-subspace alignment

to be evaluated without image generation. **Retained sites and inputs.** We retain LoRA sites in the text encoder and the key/value projections of text cross-attention in the denoising backbone [32, 35]. For a text-encoder site, $\mathbf{h}_{s,u}(x)$ is the contextual representation entering that site when processing prompt x . For a retained cross-attention site, $\mathbf{h}_{s,u}(x)$ is the final text representation supplied by the text encoder. In both cases, the required inputs are available from text conditioning alone. Query projections and other denoiser LoRA sites are excluded because their inputs depend on the evolving latent or subsequent denoising computation. **Valid LoRA updates.** For each selected site s , SGES requires both LoRA factors $\mathbf{A}_s$ and $\mathbf{B}_s$ and forms the raw update $\Delta\mathbf{W}_s = \mathbf{B}_s\mathbf{A}_s$. Sites with an incomplete factor pair or an exactly zero update are discarded, while non-finite updates are treated as invalid. For each retained site, the dominant input directions are obtained from the SVD of the unscaled update $\Delta\mathbf{W}_s$, retaining the top- k right singular directions. **Site grouping.** Retained sites are grouped by their location in the model. Text-encoder sites form the te group, while retained cross-attention sites in the down and up paths of the denoising backbone form the down and up groups, respectively. Mid-block sites are excluded in our implementation. Since an adapter may not contain retained sites in every group, we define

$$\mathcal{G}_A = \{b \in \{\text{te, down, up}\} \mid |J_b| > 0\}. \quad (35)$$

Only groups in $\mathcal{G}_A$ participate in subsequent SGES computation.

A.2 Composed-Word Discovery

A meaningful word may be split into multiple tokenizer units [33, 38], so searching only individual vocabulary entries can miss strongly aligned words. SGES therefore composes multiple lexical units into candidate words before constructing the Atomic pools. **Lexical units.** We divide alphabetic tokenizer entries into prefix units and final units according to the tokenizer’s word-ending marker. Prefix units may appear at the beginning or middle of a composed word, whereas final units terminate the word. All units are scored separately for each site group. Single-unit words are handled directly by the original vocabulary and are therefore not produced by the composer. **Progressive composition.** SGES constructs words of two and three lexical units. For each group, we first retain the top 30 prefix units and take their union across groups. Each retained prefix is paired with every final unit to form two-unit candidates. Final units are not pre-filtered by their individual scores, allowing combinations whose alignment emerges only after composition. All candidates are evaluated as complete text inputs using Equations (21) and (22), and each group retains its top 120 two-unit candidates.

Three-unit candidates are formed by inserting a middle prefix into promising two-unit compositions. The outer units are drawn from the union of the top 40 two-unit candidates

from each group, while the middle-unit set contains the top 60 prefixes from each group together with all prefixes of at most two characters. Each group retains its top 80 three-unit candidates. Composition stops after three units. **Validation and filtering.** Retained compositions are decoded and accepted only if they remain unchanged after tokenizer re-encoding and decoding. Candidates are deduplicated by decoded text, with the shorter composition retained when multiple chains produce the same word.

The valid words are then retokenized and rescored as complete candidates. In each group, a composed word must score at least as high as the strongest individual lexical unit. To suppress near-duplicate compositions, let $P(w)$ denote the set of lexical units used to form word w , and define

$$d_{\text{piece}}(w, w') = \max\{|P(w)|, |P(w')|\} - |P(w) \cap P(w')|. \quad (36)$$

Candidates are processed in descending score and retained only if $d_{\text{piece}}(w, w') > 1$ for every previously retained candidate. We additionally allow each prefix or final unit to appear in at most two retained compositions within a group.

After filtering, each group contributes at most 75 composed-word proposals. Duplicate texts across groups are merged, and composed words already represented by native whole-word vocabulary entries are removed. The remaining proposals compete directly with native words during Atomic pool construction in Appendix A.3.

A.3 Atomic Candidate Pool Construction

SGES constructs a compact Atomic pool for each present site group from two complementary candidate sources: lexical tokens and common phrases. **Candidate sources.** We consider two types of Atomics. TOKEN candidates consist of native whole-word vocabulary entries and the composed words discovered in Appendix A.2. PHRASE candidates are drawn from the fixed public prompt corpus Gustavosta/Stable-Diffusion-Prompts [14], shared across all adapters and independent of target-specific information.

To construct the phrase source, corpus prompts are lower-cased and reduced to alphabetic words of at least two characters. We extract adjacent word pairs occurring at least five times and retain those tokenized into two or three lexical units. After removing duplicate token sequences, the 3,000 most frequent phrases form the shared candidate set. Corpus frequency is used only for source construction; subsequent ranking depends solely on the LoRA-specific alignment score. **Candidate selection.** Each candidate is independently scored in every present group using $R_b(a \mid a)$ from Equation (23). TOKEN and PHRASE candidates are ranked and filtered separately. We remove non-ASCII candidates and retain only the highest-scoring candidate within each lemma-equivalent family. We also limit repeated lexical-root families to prevent closely related forms from dominating a group pool.

Because the same Atomic may rank highly in multiple groups, each candidate is assigned to the group in which it achieves the best within-type rank. Let $\text{rank}_b(a)$ denote this rank. We define

$$\text{owner}(a) = \arg \min_{b \in \mathcal{G}_A} \text{rank}_b(a), \quad (37)$$

with ties resolved by the fixed priority $\text{te} \succ \text{up} \succ \text{down}$. **Final Atomic pools.** Each present group retains the 30 highest-ranked TOKEN Atomics and the 20 highest-ranked PHRASE Atomics assigned to that group:

$$C_b = C_b^{\text{token}} \cup C_b^{\text{phrase}}, \quad |C_b^{\text{token}}| = 30, \quad |C_b^{\text{phrase}}| = 20. \quad (38)$$

Native and composed words compete for the same TOKEN quota, so composed words receive no dedicated allocation. The resulting group-specific pools provide the search space for evolutionary prompt search.

A.4 Evolutionary Search and Shortlist Construction

Although the Atomic pools substantially reduce the lexical search space, the number of prompt combinations remains large. Moreover, $\text{SAS}(p)$ depends on the jointly encoded prompt, so Atomics cannot be optimized independently. SGES therefore performs evolutionary search over complete prompts. **Evolutionary search.** Each present site group contributes three Atomic slots. Initial prompts are sampled from the corresponding group-specific pools with a mixture of rank-biased and uniform probabilities:

$$P(i) = 0.3 \frac{1}{n} + 0.7 \frac{\exp(-r_i/\tau_{\text{EA}})}{\sum_j \exp(-r_j/\tau_{\text{EA}})}, \quad \tau_{\text{EA}} = 100, \quad (39)$$

where r_i is the zero-based rank of Atomic i in a pool of size n . Sampling excludes Atomic texts already used elsewhere in the same prompt. Each run maintains a population of 48 prompts for 100 generations, and we perform three independent runs.

Prompts are ranked by contextual SAS. Parent selection uses three-way tournaments [27], followed by slot-wise crossover with probability 0.7 and per-slot mutation. Both operations preserve the site-group assignment of each slot and prevent duplicate Atomic texts. Fresh random prompts are periodically injected to maintain exploration, with a larger restart triggered when the best prompt remains unchanged for three consecutive generations. **Diversity preservation.** Maximizing SAS alone can produce many minor variations of the same high-scoring prompt. SGES therefore groups prompts according to their Atomic-set overlap. Let $S(p)$ denote the set of Atomic texts in prompt p . We define

$$d_{\text{set}}(p, p') = \max\{|S(p)|, |S(p')|\} - |S(p) \cap S(p')|. \quad (40)$$

Prompts with $d_{\text{set}}(p, p') \leq 3$ are treated as belonging to the same family, and only the highest-scoring member of each

family retains selection fitness [31]. The four strongest family representatives are preserved as elites. **Coordinate refinement.** Every three generations, SGES further refines up to four of the highest-scoring family representatives through coordinate-wise search. For each Atomic slot, all valid replacements from its group-specific pool are evaluated while the remaining slots are fixed. The replacement yielding the highest contextual SAS is retained if it improves the current prompt. Up to two passes are performed, with early termination when a complete pass produces no improvement. **Shortlist construction.** After evolution, each run retains the five highest-scoring prompts as *confirm* candidates. To increase family diversity, SGES further selects up to five *reservoir* candidates from approximately the top 5% of all evaluated prompts, requiring a distance $d_{\text{set}} > 3$ from every previously selected candidate.

SGES additionally constructs shorter candidates from at most two family-distinct high-scoring lineages. Starting from a full prompt, Atomics are greedily removed according to contextual SAS while preserving at least one Atomic from every originally present site group. We retain candidates with lengths of 7, 5, and 3 whenever feasible. Finally, candidates from all three runs are merged and deduplicated to form

$$\mathcal{P}_{\text{short}} = \{p_1, \dots, p_N\}, \quad (41)$$

which is passed to PVI for output-side concept identification.

B Evaluation Details

B.1 Benchmark and Models

We evaluate our method on 164 LoRA adapters across the SD1.5 and SDXL model families. The benchmark covers real identities, synthetic identities, anime characters, and artistic styles. It combines 48 adapters collected from public model-sharing platforms [4, 19] with 116 adapters trained locally under controlled settings.

The locally trained adapters focus on identity concepts, comprising 19 real identities and 10 synthetic identities. For each identity, we train adapters under both SD1.5 and SDXL. Within each model family, we train two matched variants that differ in how faithfully the target concept is disclosed by the trigger. The *faithful* variant uses a human-readable trigger that reveals the target identity, whereas the *unfaithful* variant uses an opaque alphanumeric trigger that does not reveal the target semantics. The matched variants use identical training images and optimization settings and differ only in the trigger used in their training captions. This construction yields 116 locally trained adapters.

Among the 164 adapters, 106 use faithful triggers and 58 use unfaithful triggers. The latter comprise 29 opaque-trigger variants in each model family. The baseline comparison, verification experiment, and main ablations use one fixed faithfully disclosed SD1.5 cohort for which every compared

method is applicable. Cross-family and trigger-disclosure results use their respective benchmark subsets, as stated with each result.

Each adapter is evaluated with a compatible base model [32, 35, 43, 44]. The benchmark includes multiple compatible checkpoints from both the SD1.5 and SDXL families.

B.2 Local LoRA Training

All locally trained identity adapters use LoRA rank $r = 128$. LoRA is applied to both the denoising network and the text encoder. For SDXL, both the CLIP-L and OpenCLIP-G text encoders are adapted [3, 32, 33, 43]. The main training configuration is summarized in Table 9.

Table 9: Training configuration for locally trained identity LoRAs.

Setting	SD1.5	SDXL
Base model	Stable Diffusion 1.5	SDXL 1.0
Training resolution	512 px	1024 px
LoRA rank	128	128
U-Net learning rate	1×10^{-4}	1×10^{-4}
Text-encoder learning rate	5×10^{-5}	5×10^{-5}
Optimizer	AdamW	AdamW
LR schedule	Constant	Constant
Training steps	3,000	1,000

We use AdamW [26] with $\beta = (0.9, 0.999)$, $\epsilon_{\text{Adam}} = 10^{-6}$, weight decay 0.01, and maximum gradient norm 1.0. Training follows the standard diffusion noise-prediction objective [15, 35]. Given an image x , a training caption p_{cap} , a sampled timestep t , and Gaussian noise ϵ , the model minimizes

$$\mathcal{L}_{\text{train}} = \mathbb{E}_{x, p_{\text{cap}}, t, \epsilon} \left[\left\| \epsilon - \epsilon_{\theta}(z_t, t, p_{\text{cap}}) \right\|_2^2 \right]. \quad (42)$$

All SD1.5 adapters are trained for 3,000 optimization steps, whereas all SDXL adapters are trained for 1,000 steps.

B.3 Training Data for Locally Trained LoRAs

The locally trained LoRAs use two sources of identity images. Real identities are represented by web-collected photographs of public figures, whereas synthetic identities are represented by images generated from fixed synthetic identity specifications. Both sources follow the same captioning protocol for LoRA training. **Real-identity images.** For each of the 19 real identities, we collect publicly available images of the corresponding public figure through web image search. The resulting training sets contain 19–46 images per identity and cover varied viewpoints, poses, clothing, scenes, and photographic conditions. **Synthetic-identity images.** For each of the 10 synthetic identities, we first construct a synthetic identity specification describing a fixed set of identity-defining

visual attributes, including face shape, eyes, hair, eyebrows, nose, jaw structure, and a distinctive feature. GPT-5.5 via Codex [30] is used to generate specifications.

Given an identity specification, we then generate a batch of images while keeping its identity-defining attributes fixed and varying only image-specific factors such as viewpoint, pose, expression, clothing, scene, lighting, framing, and shot scale. The same specification is used throughout the batch to maintain identity consistency across images. Synthetic training sets contain 25–50 images per identity. **Training captions.** Real and synthetic images follow the same captioning protocol. For each training image, GPT-5.5 via Codex [30] generates an image-specific caption describing visible factors that vary across examples, such as framing, viewpoint, pose, expression, clothing, scene, lighting, and occlusion.

Each caption begins with the trigger and a class word, uses the trigger exactly once, and contains 20–45 English words. For each matched faithful and unfaithful LoRA pair, the images and caption content are held fixed; only the identity trigger is replaced. The faithful variant uses the human-readable identity name, whereas the unfaithful variant uses the corresponding opaque alphanumeric trigger. **Generation and captioning templates.** The principal templates used for synthetic-image generation and training-caption construction are provided below. Placeholders in braces are instantiated from the fixed identity specification or trusted training meta-data.

Synthetic identity specification

Design {N} entirely synthetic human identities for a controlled diffusion-model benchmark. None should resemble or be named after a real public figure.

Make the identities visually distinguishable using independent combinations of age band, face shape, skin tone, eye shape and color, hair texture, color, and cut, eyebrow geometry, nose geometry, jaw or chin geometry, and one subtle distinctive feature.

Keep the attributes of each identity internally consistent. Avoid occupations, personality descriptions, stereotypes, costumes, fantasy features, or exaggerated deformities.

Return JSON containing:
id, synthetic_name, class_word, age_band, face_shape, skin_tone, eyes, hair, eyebrows, nose, jaw_or_chin, distinctive_feature, and

identity_signature.

Synthetic identity generation

Generate a diverse batch of photorealistic images of the same synthetic identity defined below.

IDENTITY:
{IDENTITY_SPECIFICATION}

The identity-defining attributes in the specification must remain consistent across all generated images. Every image must depict the same person.

Across the batch, vary only image-specific factors, including:
shot scale, viewpoint, pose, gaze, expression, clothing, scene, lighting, framing, and aspect ratio.

Include diverse close-up, head-and-shoulders, upper-body, three-quarter, and full-body views; front, three-quarter, and profile viewpoints; indoor and outdoor scenes; and varied natural photographic conditions.

Maintain realistic human anatomy, natural skin texture, and consistent facial identity across the entire batch. Do not introduce additional people, text, watermarks, or logos, and do not make any image resemble a real public figure.(@*)

Training caption generation

You are a conservative visual annotator creating captions for an identity-conditioned LoRA training dataset. Describe only directly visible, image-specific factors that vary across images. Do not infer the person’s identity from the image and do not repeatedly describe stable identity-defining facial attributes.

TRIGGER: "{TRIGGER}"
CLASS_WORD: "{CLASS_WORD}"

Generate a 20-45-word English caption that:
1. begins with the exact TRIGGER followed by

CLASS_WORD;
2. uses TRIGGER exactly once;
3. describes only visible factors such as shot scale, viewpoint, pose, gaze, expression, clothing, scene, lighting, occlusion, and rendering;
4. contains no additional proper name or inferred identity.

B.4 Baseline Configurations

We compare with six prior methods from two related problem settings: concept-agnostic difference discovery and concept-specified verification. Table 10 summarizes how each method is used in our LoRA setting and the concept information it requires. We use official implementations when available [7, 9, 21, 49, 51]. Since PAIA does not provide public code, we independently reproduce it from the published method.

Table 10: Prior methods, their use in our LoRA evaluation, and the concept information they require.

Method	Use in LoRA evaluation	Required input
DriftScope	Rank tokens by their LoRA–base drift	None
CompCon	Search for prompts exposing LoRA–base differences	None
VisDiff	Describe differences between paired LoRA and base image sets	None
DPO-Diff-EPO	Optimize prompts for paired-model divergence	None
PEZ	Recover a hard prompt from concept images	Concept images
PAIA	Verify supplied concepts for each LoRA	Candidate images

For every prompt-returning baseline, the reported carrier is selected using that method’s own score or ranking; target-concept references are used only for final evaluation and never for carrier selection. We retain each method’s standard search configuration and use the same generation and final-evaluation protocol wherever the method’s task permits. PEZ receives a concept-image set that is disjoint from the held-out references used to compute target-concept discrepancy. **Concept-agnostic difference discovery.** These methods do not assume a concept to be examined in advance. Instead, we adapt them to expose changes introduced by the LoRA through model comparison, output comparison, or prompt optimization.

DriftScope [22] studies semantic drift introduced by diffusion-model adaptation. Given an original and an adapted

model, it identifies lexical tokens whose associated visual concepts change most after adaptation. We instantiate the model pair with the compatible base and LoRA-enabled models and insert the resulting high-drift tokens into DriftScope’s prompt templates to form candidate prompts.

CompCon [8] compares two T2I models and discovers visual attributes on which their generations diverge, together with prompts that expose these differences. We use the LoRA-enabled and compatible base models as the model pair. Starting from generic prompts, CompCon identifies LoRA–base visual differences and searches for prompts that reliably expose them. The resulting verified prompts are used as candidate prompts.

VisDiff [10] describes visual differences between two fixed image sets. We construct one set from LoRA-enabled generations and the other from compatible base-model generations using the same generic prompts and random seeds. VisDiff produces natural-language descriptions that best distinguish the two sets, which we directly use as candidate prompts. Unlike CompCon, VisDiff describes differences in fixed image sets rather than searching for new prompts that expose them.

DPO-Diff-EPO [50] performs discrete prompt optimization for diffusion models. We retain its EPO optimizer but replace the original task objective with one that favors visual divergence between paired LoRA-enabled and compatible base-model generations. The resulting prompts therefore expose generation behavior changed by the LoRA.

Concept-specified verification. These methods require the concept to be specified in advance. PEZ recovers prompts from concept examples and provides indirect evidence of whether the concept can be elicited, whereas PAIA directly verifies whether the specified concept is encoded.

PEZ [52] performs image-guided hard-prompt optimization. Given concept images, it optimizes a prompt in continuous embedding space and projects it onto discrete vocabulary tokens. In our evaluation, PEZ accesses neither the LoRA adapter under evaluation, its compatible base model, nor its training trigger; it receives only the supplied concept images. The recovered prompt is subsequently evaluated on the LoRA-enabled model to determine whether it elicits the specified concept.

PAIA [55] is the closest prior work in terms of problem setting. It verifies whether a LoRA-adapted diffusion model has learned a prespecified concept by comparing its denoising behavior with that of the compatible base model. Given images of the candidate and irrelevant concepts, it returns a binary learned/not-learned decision. We query each candidate concept independently using its concept images. PAIA therefore verifies supplied concepts rather than discovering an unknown concept or producing a carrier prompt. Since no official implementation is publicly available, we independently reproduce the published method.

C Additional Sensitivity Analysis

C.1 Retained Subspace Dimension

Every value of k reconstructs the Atomic pools and reruns the complete discovery pipeline rather than reusing candidates from the default configuration.

Table 11: Effect of the maximum number of leading LoRA input directions retained per site. Lower target-concept discrepancy is better.

k	Real ID ↓	Synthetic ID ↓	Anime ↓	Mean ↓
1	0.080	0.202	0.144	0.128
2	0.078	0.178	0.123	0.114
4	0.088	0.181	0.144	0.127

For site s , the number of retained directions is $k_s = \min(k, \rho_s)$, where $\rho_s = \text{rank}(\Delta \mathbf{W}_s)$. Table 11 varies k while selecting the leading directions. Retaining two directions performs best, and the mean varies by at most 0.014 over the tested values. With $k = 1$, real-identity performance is nearly unchanged (0.080 versus 0.078), whereas synthetic identities and anime characters degrade more noticeably. Increasing to $k = 4$ primarily hurts real identities and anime. This pattern is consistent with one direction sometimes omitting complementary responses and four directions admitting weaker ones. The experiment therefore shows limited sensitivity within $k \in \{1, 2, 4\}$, but does not establish a universal optimum outside that range.

D Adaptive Attack Details

D.1 Threat Model and Controls

The adaptive creator knows that SCOPE uses leading LoRA input directions, but not the number of retained directions, the selected site groups, or the remaining hyperparameters used by SCOPE. The creator can modify LoRA training but must preserve the LoRA-enabled model’s ability to generate the adapter’s dominant hidden concept. In this controlled evaluation, utility is measured against the target concept c .

We apply the attack to the controlled faithful SD1.5 identity adapters. Each attacked adapter uses exactly the same training images, captions, human-readable trigger, compatible base model, LoRA target modules, rank and scaling factor, optimization steps, optimizer, learning rates, scheduler, batch configuration, resolution, and seed policy as its matched clean adapter. No LoRA module is added, removed, or moved. The sole change is the regularizer described below, applied to every LoRA module already trained by the original recipe.

D.2 Subspace-Dispersion Objective

For a nonzero trained LoRA module s , let

$$\Delta \mathbf{W}_s = \mathbf{B}_s \mathbf{A}_s \quad (43)$$

have singular values $\{\sigma_{s,j}\}$. We define its scale-invariant spectral concentration as

$$\text{SC}_s = \frac{\sum_j \sigma_{s,j}^4}{\left(\sum_j \sigma_{s,j}^2\right)^2}. \quad (44)$$

A spectrum concentrated in a few directions has a larger SC_s , whereas a flatter spectrum has a smaller value. Let $\mathcal{M}_{\text{LoRA}}$ contain every LoRA module in the clean training recipe. The attack minimizes

$$\mathcal{R}_{\text{disp}} = \frac{1}{|\mathcal{M}_{\text{LoRA}}|} \sum_{s \in \mathcal{M}_{\text{LoRA}}} \text{SC}_s. \quad (45)$$

The spectral concentration reported in Section 5.4.2 is this module-wise mean.

The regularizer is computed without forming the full update matrix. Let

$$\mathbf{G}_{A,s} = \mathbf{A}_s \mathbf{A}_s^\top, \quad \mathbf{G}_{B,s} = \mathbf{B}_s^\top \mathbf{B}_s. \quad (46)$$

Then the same concentration can be computed exactly as

$$\text{SC}_s = \frac{\text{tr}(\mathbf{G}_{A,s} \mathbf{G}_{B,s} \mathbf{G}_{A,s} \mathbf{G}_{B,s})}{[\text{tr}(\mathbf{G}_{A,s} \mathbf{G}_{B,s})]^2}. \quad (47)$$

This computation uses only rank-dimensional matrices and preserves gradients. The Gram matrices, concentration, and dispersion loss are computed in FP32; the original mixed-precision diffusion training is otherwise unchanged.

D.3 Fixed Training Schedule

Let v denote the optimization step. The attacked training objective is

$$\mathcal{L}_{\text{atk}}(v) = \mathcal{L}_{\text{train}} + \lambda_{\text{disp}}(v) \mathcal{R}_{\text{disp}}, \quad (48)$$

where the attack strength is fixed rather than selected using SCOPE’s discovery results:

$$\lambda_{\text{disp}}(v) = \begin{cases} 0, & v < 300, \\ 0.1(v - 300)/300, & 300 \leq v < 600, \\ 0.1, & v \geq 600. \end{cases} \quad (49)$$

The initial warm-up avoids evaluating the concentration ratio when the LoRA update is near zero. The coefficient is not searched or adjusted after training.

D.4 Evaluation Protocol

Section 5.4.2 defines utility preservation and conditional evasion. Each matched clean–attacked pair uses the same fixed p_{trig} containing the human-readable training trigger; it is set before attack training and unavailable to the moderator. We render p_{trig} and the carrier prompt p^* with the attacked LoRA-enabled model $\mathcal{B}_{\mathcal{A}}$ and evaluate both against the same attack-independent held-out references for the target concept c . Training uses neither the references nor discovery results.