This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

TriXfer: Triple-Cross Transfer Attack on Vision Transformer–Adapted Downstream Classifiers

Mengdie Huang, Yingjun Lin, Ninghui Li, *Fellow, IEEE*, Elisa Bertino, *Fellow, IEEE*

***Abstract*—Vision Transformer (ViT)–based foundation models are widely adopted for downstream image classification due to their highly transferable representations. However, reliance on pretrained ViTs also introduces security risks, as adversarial vulnerabilities may propagate from the pretrained model to downstream classifiers. Existing studies largely examine this threat under strong assumptions that are rarely satisfied in practice, such as white-box access to the pretrained model and the ability to craft adversarial examples (AEs) on clean downstream data. In this work, we investigate a significantly stricter and previously unexplored threat model in which both the pretrained ViT and downstream classifiers are accessible only as black boxes, and attacks rely solely on a surrogate model and dataset with a mismatched label space. We address two fundamental questions: (i) whether downstream classifiers remain vulnerable to AEs crafted against a black-box pretrained model, and (ii) whether downstream robustness is primarily determined by the pretraining mechanism or by the downstream adaptation process. To answer these questions, we propose `TriXfer`, the first Untargeted Triple-Cross Transfer Attack framework that enables AEs generated on clean surrogate data to transfer simultaneously across model architectures, datasets, and label spaces. `TriXfer` aligns mismatched label spaces at the semantic concept level and jointly optimizes adversarial perturbations using task-level, concept-level, and hidden-representation objectives to enhance transferability. Extensive experiments across a wide range of ViT backbones, spanning supervised, self-supervised, and vision-language paradigms, alongside standard and adversarial pretraining, and diverse downstream adaptation strategies demonstrate that `TriXfer` is more effective in compromising both pretrained ViTs and downstream classifiers than baseline attacks using diverse surrogate models and datasets, revealing persistent inherited black-box vulnerabilities.**

***Index Terms*—Vision transformer, downstream adaptation, transfer attack, adversarial example, image classification.**

I. Introduction

Downstream image classifiers adapted from pretrained Vision Transformer (ViT) family-based foundation models [1] achieve substantial gains, owing to the ability of transformer backbones [2] to learn rich, transferable representations from large-scale pretraining data [3]–[7]. However, the widespread adoption of ViT-based transfer learning (TL) pipelines has raised security concerns, as the feature representation shared between a pretrained ViT and its adapted downstream models facilitates transfer attacks, a black-box attack in which adversarial examples (AEs) crafted on a white-box model mislead unseen black-box victims [8]–[10]. Prior work shows that ViTs are vulnerable not only to ViT-specific standard [11] and adaptive [12], [13] white-box attacks, but also to ViT-tailored cross-architecture transfer attacks [10], [14]–[18]. Moreover, recent studies [19]–[21] show that downstream classifiers adapted from pretrained ViTs [3], [22]–[25] via various TL strategies (e.g., full fine-tuning, linear probing, and parameter-efficient tuning [26]–[28]) can be misled by transferable AEs crafted on clean downstream data using the white-box pretrained ViT, indicating that white-box vulnerabilities in pretrained ViTs can propagate to black-box downstream models.

The above findings, however, do not necessarily imply that downstream classifiers remain vulnerable to transfer attacks under more restrictive and realistic threat models, in which neither the pretrained ViT backbone nor the downstream model is accessible to the attacker. In this dual black-box scenario, the attacker can no longer exploit the exact shared representations or architectural similarity between the pretrained model and its downstream classifier, and must instead rely on a surrogate model whose architecture, parameters, training data, or even label space may differ substantially from those of the victim models. Thus, it remains unclear whether black-box adversarial vulnerabilities in pretrained ViTs against transfer attacks can be inherited by downstream models.

This gap motivates us to study two essential and previously unexplored questions: ① *Do black-box downstream classifiers remain vulnerable to transferable AEs that can successfully mislead a black-box pretrained ViT?* Answering this question is crucial for assessing whether concealing the pretrained backbone can meaningfully improve downstream robustness. If downstream classifiers are indeed vulnerable in such black-box settings, a deeper issue remains unresolved. ② *Is downstream robustness primarily determined by the pretraining mechanism or by the downstream adaptation strategy?* Understanding the relative contributions of these two factors is fundamental for determining where future defensive efforts should be prioritized, namely whether robustness should be enforced at pretraining or during downstream adaptation.

Although several publicly available pretrained models can be used by an attacker as surrogate models without requiring extensive surrogate training from scratch, such surrogates are rarely identical to either the victim pretrained ViT or its downstream classifier. In practice, three types of mismatch commonly arise between the surrogate and the victim (see Table I): (i) Model mismatch, due to differences in architecture and parameters; (ii) Dataset mismatch, caused by discrepancies between surrogate and victim data domains; and (iii) Label space mismatch, stemming from differences in semantic granularity

Manuscript received 30 January 2026. Revised 19 June 2026. *Corresponding author: Elisa Bertino*.

Mengdie Huang, Yingjun Lin, Ninghui Li, and Elisa Bertino are with the Department of Computer Science, Purdue University, West Lafayette 47906, United States (e-mail: huan1932@purdue.edu; lin1368@purdue.edu; ninghui@purdue.edu; bertino@purdue.edu).

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

TABLE I
DIMENSIONS OF TRANSFER MISMATCH IN TRANSFER ATTACKS

Dimensions	Definition
Cross-Model	Attacks crafted on one model transfer to a victim model with a different architecture.
Cross-Dataset	Attacks crafted on one dataset transfer to a victim model trained on a different dataset.
Cross-LabelSpace	Attacks crafted on one label space transfer to a victim model with a different label space in class hierarchy.

and class hierarchy. The coexistence of these *cross-model*, *cross-dataset*, and *cross-label space* mismatches makes it non-trivial for AEs crafted using a triple-cross surrogate model to reliably transfer to both the pretrained ViT and downstream classifiers under a dual black-box setting. Moreover, existing transfer attacks typically assume access to target-domain data (e.g., clean downstream samples). However, this assumption may not hold in many vision foundation model–based TL pipelines, where a pretrained ViT is adapted to diverse downstream tasks using private or proprietary datasets that are unavailable to the attacker. This further motivates us, for the first time, to study transferability under a stricter threat model in which AEs are crafted entirely from clean surrogate data.

Our Work. In this paper, we propose `TriXfer`, the first untargeted triple-cross transfer attack designed to evaluate adversarial transferability in downstream adaptation (see Fig. 1). `TriXfer` enables AEs to be crafted using a surrogate model and clean surrogate data, while transferring simultaneously across model architectures, datasets, and label spaces to attack a black-box pretrained ViT and its downstream classifiers. To overcome label-space differences between surrogate and victim models, `TriXfer` performs label semantic alignment, allowing perturbations to be optimized with respect to overlapped semantic concepts in addition to exact class labels. Also, we design a novel multi-level attack objective that jointly optimizes surrogate task-level, semantic concept-level, and hidden representation-level losses, enhancing transferability in the triple-cross scenario. Our method operates under minimal attacker assumptions, where neither the pretrained backbone nor the downstream models, nor their datasets, are accessible.

Evaluation. We conduct extensive experiments to evaluate `TriXfer` and answer two research questions posed. Our evaluation spans pretrained ViT backbones under supervised (ViT-B/16 [3], Swin-B/P4W7 [29], CaiT-B [24], DeiT III-B [22], TinyViT [25]), self-supervised (DINOv2 ViT-S [23]), vision-language (CLIP ViT-B/16 and CLIP ViT-L/14), and adversarial (AT-ViT-B/16 and AT-Swin-B/P4W7 [30]) pretraining, together with downstream adaptation strategies including fine-tuning, linear probing, adapter tuning [26], LoRA [27], prompt tuning, and prefix tuning [28]. Surrogate models include CNN-based (ConvNeXt V2 [7], InceptionNeXt [31]) and lightweight ViT-based (MobileViT [32], EfficientViT [33]) architectures, fine-tuned on surrogate datasets (Oxford-IIIT Pets-37 [34], Stanford Cars-196 [35], FGVC-Aircraft-100 [36], ImageWoof-10 [37], STL-10 [38], and CINIC-10 [39]) that differ from the victim downstream datasets (CIFAR-10, CIFAR-100 [40], and ImageNette [41]) in data domain and label granularity.

We divide `TriXfer` attack modes by the surrogate–target label-space relationship: (i) *Hierarchical semantic overlap*, where surrogate and target labels lie at different levels of a semantic hierarchy, such as fine-grained surrogate subcategories map to coarse-grained target categories (e.g., Oxford-IIIT Pets-37, Stanford Cars-196, FGVC-Aircraft-100, or ImageWoof-10→CIFAR-10); (ii) *Non-hierarchical semantic overlap*, where surrogate and target share label sets at the same or similar semantic level but originate from different semantic sources (e.g., CINIC-10 or STL-10→CIFAR-10). Against leading transfer attack baselines (SP [9], cross-domain UAP [42], and XTransfer [43]), each combined with five perturbation operators (FGSM [44], IFGSM [45], MIFGSM [46], DIFGSM [47], and TIFGSM [48]), `TriXfer` achieves superior downstream attack performance on the standard attack success rate (ASR) over all AEs and our proposed conditional attack success rate (ConASR) over only the AEs that first fool the pretrained ViT-based foundation model, revealing inherited black-box vulnerabilities in downstream classifiers.

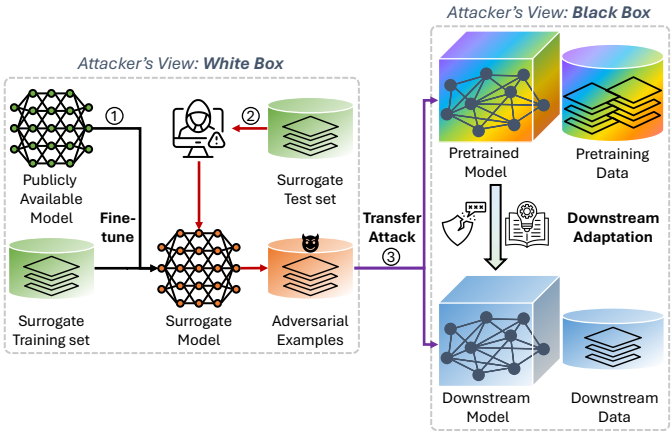


Fig. 1. Workflow of the `TriXfer` attack in downstream adaptation. ① The attacker efficiently fine-tunes a public model on the surrogate training set to obtain a surrogate model. ② Using this surrogate model and clean surrogate data, the attacker generates AEs. ③ AEs that successfully fool the black-box pretrained ViT are then used to attack the black-box downstream models.

In summary, we make the following contributions:

- We are the first to study the transferability of black-box adversarial vulnerabilities from pretrained ViTs to downstream classifiers, where both the pretrained ViT backbone and the downstream classifier are black boxes, and AEs are crafted from clean surrogate data on a lightweight surrogate.
- We propose `TriXfer`, the first untargeted triple-cross transfer attack, integrating LLM-based label semantic alignment, hierarchical surrogate fine-tuning, and a multi-level attack objective, enabling AEs to transfer simultaneously across architectures, datasets, and label spaces.
- We integrate `TriXfer` with five basic perturbation operators and evaluate it across diverse ViTs, adaptation strategies, downstream datasets, surrogate models, and surrogate datasets. Extensive experiments show that `TriXfer` consistently outperforms attack baselines and, under ConASR, that black-box vulnerabilities in pretrained ViTs can propagate to downstream classifiers. We also conduct a class-wise analysis of transferability across different label-alignment levels, and show the attack is robust to the stochasticity and cross-LLM disagreement of the LLM-based label alignment.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

II. Background

A. Vision Transformer

A ViT processes an image through token-based self-attention (see Fig. 2). An input $x \in \mathbb{R}^{H\times W\times C}$ is split into $N = HW/P^2$ patches $\{x_p^i\}_{i=1}^N$ of size $P \times P$, each linearly projected into a D-dimensional token embedding $z_0^i = Ex^i$ via a learnable matrix E. A learnable class token $z_0^0 = [\texttt{CLS}]$ is prepended and positional embeddings E_{pos} are added, forming the first encoder input $z_0 = [z_0^0, z_0^1, \ldots, z_0^N] + E_{pos}$. Stacked Transformer encoder blocks then apply multi-head self-attention to capture global, long-range dependencies, and the final $[\texttt{CLS}]$ embedding is fed to the classification head.

Transformer Encoder. Each block $l \in \{1, \ldots, L\}$ applies multi-head self-attention (MSA) and an MLP, each wrapped with LayerNorm (LN) and a residual connection. MSA derives queries, keys, and values from the normalized input via learnable projections W_Q, W_K, W_V, as in (1).

$$Q = \mathrm{LN}(z_{l-1})W_Q,\ K = \mathrm{LN}(z_{l-1})W_K,\ V = \mathrm{LN}(z_{l-1})W_V. \tag{1}$$

MSA computes k parallel attention heads $\{head_i\}_{i=1}^k$, each performing scaled dot-product attention as defined in (2).

$$\mathrm{head}_i = \mathrm{softmax}(Q_i K_i^\top/\sqrt{d_k})\, V_i, \tag{2}$$

where d_k is the per-head dimension. The k heads are concatenated and projected by W_O, and the block output follows from two residuals, $\tilde{z}_l = \mathrm{MSA}(\mathrm{LN}(z_{l-1})) + z_{l-1}$ and $z_l = \mathrm{MLP}(\mathrm{LN}(\tilde{z}_l)) + \tilde{z}_l$.

Classifier Head. After L blocks, the classification prediction is computed solely from the final class token $z_L^0 = [\texttt{CLS}]$ via a learnable projection $W_{cls} \in \mathbb{R}^{D\times C}$, as in (3).

$$y = \mathrm{softmax}(\mathrm{LN}(z_L^0)W_{cls}). \tag{3}$$

B. Downstream Adaptation

Downstream adaptation is a form of transfer learning (TL), in which a model pretrained on a large dataset is subsequently adapted to a downstream task using downstream data.

Data Distribution. We adopt the standard TL assumption [49] that both *pretraining-domain* samples x^P and downstream-domain (i.e., *target-domain*) samples x^T are drawn from the same marginal distribution $\mathbb{P}_X$ over a shared input space $\mathcal{X}$. This assumption implies that a shared feature representation is meaningful across pretraining and downstream tasks. The downstream model (i.e., *target model*) F^T, derived by adapting a *pretrained model* F^P to the target dataset D^T via a TL algorithm $\mathcal{T}$, i.e., $F^T = \mathcal{T}(F^P, D^T)$, maps $x^T \in \mathcal{X}$ to the target-domain label space $\mathcal{Y}^T$.

The large-scale pretraining dataset D^P consists of data pairs (x^P, y^P), which are sampled i.i.d. from a joint distribution $\mathbb{P}^P$ supported on $\mathcal{X}$ and the pretraining-domain label space $\mathcal{Y}^P$. Pretraining distribution $\mathbb{P}^P$ factors as (4).

$$\begin{aligned}\mathbb{P}^P(x^P, y^P) &= \mathbb{P}^P_{f^P\circ h}(x^P, y^P)\\ &= \mathbb{P}_X(x^P)\ \mathbb{P}_{Y^P|X}\left(y^P \mid f^P \circ h(x^P)\right),\end{aligned} \tag{4}$$

where $\mathbb{P}_{Y^P|X}$ denotes the conditional distribution of pretraining labels given features produced by $f^P \circ h$, with $h : \mathcal{X} \to \mathbb{R}^r$ being the shared feature representation mapping and $f^P : \mathbb{R}^r \to \mathbb{R}$ the pretraining task–specific feature mapping.

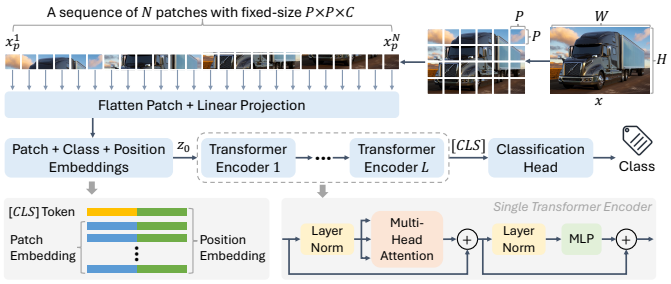


Fig. 2. The ViT pipeline for image classification, including patch embedding, stacked Transformer encoder blocks, and a class-token classification head.

The small-scale target-domain dataset D^T consists of the data pairs (x^T, y^T) sampled i.i.d. from a joint distribution $\mathbb{P}^T$ supported on $\mathcal{X} \times \mathcal{Y}^T$. Target distribution $\mathbb{P}^T$ factors as (5).

$$\begin{aligned}\mathbb{P}^T(x^T, y^T) &= \mathbb{P}^T_{f^T\circ h}(x^T, y^T)\\ &= \mathbb{P}_X(x^T)\mathbb{P}_{Y^T|X}(y^T|f^T \circ h(x^T)),\end{aligned} \tag{5}$$

where $\mathbb{P}_{Y^T|X}$ is the conditional distribution of the downstream labels given the features produced by $f^T \circ h$, and $f^T : \mathbb{R}^r \to \mathbb{R}$ is the downstream task-specific feature mapping function.

Adaptation Strategy. Let θ^P and θ^T denote the parameters of F^P and F^T. We consider the following strategies.

(i) Parameter fine-tuning adapts F^P by updating all or a subset of θ^P under target-task supervision, as in (6), covering full fine-tuning (all parameters) and linear probing (head only).

$$\min_{\theta^P} \mathbb{E}_{(x^T,y^T)\sim D^T}[\mathcal{L}(F^P(x^T;\theta^P), y^T)], \tag{6}$$

(ii) Parameter-efficient adaptation, as in (7), freezes θ^P and learns a set of task-specific parameters ϕ, giving $\theta^T = (\theta^P, \phi)$.

$$\min_{\phi} \mathbb{E}_{(x^T,y^T)\sim D^T}\Big[\mathcal{L}(F^P(x^T;\theta^P,\phi), y^T)\Big], \tag{7}$$

Common forms of ϕ are as follows. Adapters insert residual bottlenecks into Transformer blocks, $z_l \leftarrow z_l + W_l^\uparrow \sigma(W_l^\downarrow z_l)$, with down-/up-projection matrices $\phi = \{W_l^\downarrow, W_l^\uparrow\}$. LoRA applies low-rank updates to MSA projections (e.g., $W_V \leftarrow W_V + B_V A_V$), with $\phi = \{A_V, B_V\}$. Prompt tuning prepends m learnable prompt tokens $\phi = \{p_i\}_{i=1}^m$ to the input sequence, replacing z_0 with $[p_1, \ldots, p_m, z_0]$. Prefix tuning prepends (key, value) pairs $\phi = \{K_p, V_p\}$ to the attention mechanism, updating $\mathrm{head}(Q, K, V) \leftarrow \mathrm{head}(Q, [K; K_p], [V; V_p])$.

C. Transfer Attack

When the attacker has no access to the black-box victim models (F^T, F^P), a transfer attack crafts adversarial examples $x^{S*} = x^S + \delta$ on a white-box *surrogate model* F^S. The attacker samples clean inputs $x^S \sim \mathbb{P}_X$ from the same marginal distribution $\mathbb{P}_X$ as the victim domain (e.g., D^T, D^P). Surrogate dataset D^S contains pairs (x^S, y^S) drawn i.i.d. from a joint distribution $\mathbb{P}^S$ over $\mathcal{X}$ and surrogate label space $\mathcal{Y}^S$. In an untargeted attack, perturbation δ is computed by maximizing the loss of F^S, as in (8), where y^S is the ground-truth label of x^S and ϵ is the budget under the L_p norm.

$$\delta = \arg\max_{\|\delta\|_p \le \epsilon} \mathcal{L}(F^S(x^S + \delta), y^S). \tag{8}$$

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

III. Adversary Model and Problem Scope

We study black-box attacks based on transferable adversarial examples (AEs) against pretrained ViT foundation models and their downstream classifiers in image classification.

Adversary Model. The goal of our transfer attack is to generate AEs from clean surrogate data using a surrogate model F^S and to evaluate whether these AEs simultaneously fool the pretrained model F^P and its downstream model F^T. The attacker has no access to the architectures, parameters, gradients, training or test datasets of the black-box F^P and F^T, observing only hard-label outputs. The attacker relies on a white-box surrogate model F^S, obtained by fine-tuning a public model on a surrogate dataset D^S, and crafts transferable AEs from clean test samples in D^S using gradients from F^S.

This adversary model differs from prior work on transfer attacks in downstream adaptation, which assumes white-box access to F^P and generates AEs from clean downstream data. We consider a more realistic and restrictive scenario where both F^P and F^T are black boxes, and the attacker can only obtain D^S sampled from the same marginal distribution $\mathbb{P}_X$ as datasets D^P and D^T. The surrogate dataset D^S consists of data pairs (x^S, y^S), which are sampled i.i.d from a joint distribution $\mathbb{P}^S$ supported on $\mathcal{X}$ and the surrogate-domain label space $\mathcal{Y}^S$. Surrogate distribution $\mathbb{P}^S$ factors as (9).

$$\begin{aligned}\mathbb{P}^S(x^S, y^S) &= \mathbb{P}^S_{f^S \circ h}(x^S, y^S) \\ &= \mathbb{P}_X(x^S)\mathbb{P}_{Y^S|X}(y^S | f^S \circ h(x^T)),\end{aligned} \tag{9}$$

where $\mathbb{P}_{Y^S|X}$ is the conditional distribution of the surrogate labels given the features produced by $f^S \circ h$, and $f^S : \mathbb{R}^r \to \mathbb{R}$ is the surrogate task-specific feature mapping function. While the attacker does not know the exact D^P and D^T, it is realistic to assume knowledge of the data modality and label semantic scope of the victim task (e.g., image classification over animal species or vehicles), as such information is routinely disclosed in model documentation or inferred from black-box queries. This does not require knowing which specific dataset the victim uses. Any surrogate dataset whose label space conceptually overlaps with the victim semantic scope suffices. This allows the attacker to collect surrogate data that aligns with $\mathbb{P}_X$, even though it may differ substantially from the true D^P and D^T in content, scale, and label semantics.

Definition 1 (Transfer Attack Game under Downstream Adaptation)**.** *The game is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. $\mathcal{C}$ has access to F^P, D^T, and $\mathbb{P}_X$. $\mathcal{A}$ has access to F^S and black-box query access to F^P and F^T.*

1. *(Downstream Adaptation) $\mathcal{C}$ generates a downstream model $F^T \leftarrow \mathcal{T}(F^P, D^T)$ and grants $\mathcal{A}$ query access to F^T.*
2. *(Challenge Sample) $\mathcal{C}$ samples an input x from $\mathbb{P}_X$ with ground-truth labels $y^P \in \mathcal{Y}^P$ and $y^T \in \mathcal{Y}^T$ in the pretraining and downstream label spaces, respectively, such that y^P and y^T correspond to overlapping semantic concepts. $\mathcal{C}$ provides $\mathcal{A}$ with input tuple (x, y^P, y^T), norm constraint $\|\cdot\|_p$, budget $\varepsilon > 0$, and query access to F^P and F^T.*
3. *(Adversary Output) $\mathcal{A}$ returns an adversarial example $x^* = x+\delta$ generated using F^S, subject to perturbation $\|\delta\|_p \le \varepsilon$.*
4. *(Success Condition) $\mathcal{A}$ succeeds if x^* is misclassified by both victim models, i.e., $F^P(x^*) \neq y^P$ and $F^T(x^*) \neq y^T$.*

Problem Scope. To address two fundamental and previously unexplored research questions: (i) *whether downstream classifiers remain vulnerable to AEs crafted against a black-box pretrained model*, and (ii) *whether downstream robustness is mainly influenced by the pretraining mechanism or the downstream adaptation process*. We generate AEs from clean test samples in D^S using F^S and evaluate their transferability to both F^P and F^T. Due to the dual black-box attack setting, F^S and D^S may differ from the victim models (F^P and F^T) and victim domains (D^P and D^T) along multiple dimensions. Motivated by this practical constraint, we study a novel triple-cross transfer attack, in which mismatches between the surrogate and victim arise simultaneously in three aspects:

- Model mismatch: F^S may differ from the victim model in model family, architectural components, and network depth.
- Dataset mismatch: D^S may differ from the victim domain in dataset scale, visual diversity, and data collection process.
- Label Space mismatch: $\mathcal{Y}^S$ may differ from the victim label space in semantic granularity and label hierarchy.

The coexistence of these discrepancies makes transfer attack more challenging than conventional cross-model or cross-dataset attacks. In particular, the lack of aligned label spaces prevents direct optimization of label-level attack objectives and necessitates alignment at a higher-level semantic concept. We formalize, for the first time, a transfer attack game (see Definition 1) under downstream adaptation between an adversary and a challenger in the dual black-box setting.

IV. TriXfer Attack

We present `TriXfer`, which crafts AEs on a surrogate model and data that transfer to a black-box pretrained ViT and its downstream model, despite model, dataset, and label-space mismatches between the surrogate and the victims.

A. Attack Method

Acquisition of Overlapping Concept Space. Pretrained model F^P and downstream model F^T can be viewed as operating on a shared latent representation mapping h (see (4)–(5)), with task-specific feature mappings introduced on top of h. Thus, although the classes in the pretraining-domain and target-domain label spaces $\mathcal{Y}^P$ and $\mathcal{Y}^T$ may differ, the two label spaces are expected to exhibit semantic concept overlap. This overlap allows the same input samples to be meaningfully evaluated under both F^P and F^T, since samples drawn from the intersection of $\mathcal{Y}^P$ and $\mathcal{Y}^T$ can be assigned semantically valid labels in both label spaces, thereby enabling the same transferable AEs to attack F^P and F^T simultaneously.

However, due to differences in label vocabularies and semantic granularity between $\mathcal{Y}^P$ and $\mathcal{Y}^T$, directly comparing them based on surface forms (e.g., label names or IDs) is insufficient for identifying their overlap. To this end, publicly available off-the-shelf large language models (LLMs) are utilized as external semantic tools to construct semantic mappings, $\phi_P : \mathcal{Y}^P \to \mathcal{S}$ and $\phi_T : \mathcal{Y}^T \to \mathcal{S}$, where $\mathcal{S}$ denotes a concept space at a consistent level of semantic abstraction. The overlapping concept space $\mathcal{S}_\cap$ is then defined as (10).

$$\mathcal{S}_\cap = \phi_P(\mathcal{Y}^P) \cap \phi_T(\mathcal{Y}^T). \tag{10}$$

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

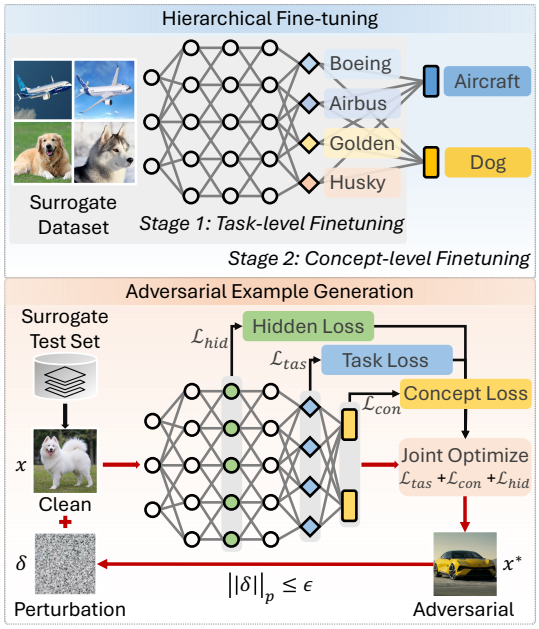


Fig. 3. Surrogate training and adversarial example generation in TriXfer.

Accordingly, the corresponding overlapping label subsets for F^P and F^T are given by (11).

$$\mathcal{Y}^P_\cap = \{y^P \in \mathcal{Y}^P : \phi_P(y^P) \in \mathcal{S}_\cap\},$$
$$\mathcal{Y}^T_\cap = \{y^T \in \mathcal{Y}^T : \phi_T(y^T) \in \mathcal{S}_\cap\}. \tag{11}$$

This procedure enables the attacker to construct the semantic intersection of $\mathcal{Y}^P$ and $\mathcal{Y}^T$ using black-box queries and public LLMs. In particular, TriXfer remains effective even without label-level alignment by operating at the semantic concept level, thereby enabling transfer attacks across label spaces. Although this concept mapping is produced by external LLMs, we show in § V-H that TriXfer's effectiveness is robust to the stochasticity of LLMs and cross-LLM disagreement.

Construction of Surrogate Dataset. Given the overlapping concept space $\mathcal{S}_\cap$ between $\mathcal{Y}^P$ and $\mathcal{Y}^T$, the attacker leverages this information to construct a surrogate dataset D^S and a surrogate model F^S for transfer attacks. Let $D^S_o = \{(x^S, y^S)\}$ denote an initial collection of publicly available labeled samples drawn from a surrogate data distribution $\mathbb{P}^S$. The surrogate dataset D_S, which will be used to train the surrogate model F^S, is constructed by filtering D^S_o to retain only those samples whose labels map into $\mathcal{S}_\cap$. Formally, the surrogate dataset is defined as $D_S = \{(x^S, y^S) \in D^S_o : \phi_S(y^S) \in \mathcal{S}_\cap\}$, where $\phi_S : \mathcal{Y}_S \to \mathcal{S}$ maps surrogate labels to the semantic concept space $\mathcal{S}$ through the off-the-shelf LLM. This mapping is equivalent to relabeling the surrogate data with concept labels aligned with the victim. Unlike relabeling based directly on victim labels, operating in the shared concept space $\mathcal{S}_\cap$ naturally accommodates one-to-many and abstract label alignments, and enables the same AEs to transfer simultaneously to both F^P and F^T. Each surrogate sample $(x^S, y^S) \in D_S$ admits semantically valid label interpretations under both F^P and F^T. Specifically, there exist corresponding ground-truth labels (y^P, y^T) in the label spaces of F^P and F^T such that semantic consistency holds at the concept level, i.e., $\phi_P(y^P) = \phi_T(y^T) = \phi_S(y^S)$.

Hierarchical Fine-tuning of Surrogate Model. The attacker initializes a surrogate model F^S from a publicly available pretrained model, which may differ from the victims F^P and F^T in architecture, and fine-tunes it on D_S as in (12).

$$\min_\theta \mathbb{E}_{(x^S,y^S)\sim D_S}[\mathcal{L}(F^S(x^S;\theta), y^S)] \tag{12}$$

To emphasize semantic alignment with $\mathcal{S}_\cap$, a concept-level classifier head $\tilde{F}^S$ parameterized by ψ is appended to the fine-tuned F^S (see Fig. 3) and optimized with the backbone frozen, as in (13), where $\phi_S(y^S) \in \mathcal{S}_\cap$ is the true concept of y^S.

$$\min_\psi \mathbb{E}_{(x^S,y^S)\sim D_S}[\mathcal{L}(\tilde{F}^S(F^S(x^S);\psi), \phi_S(y^S))], \tag{13}$$

This two-stage hierarchical fine-tuning (see Algorithm 1) first fine-tunes F^S on surrogate hard labels y^S to learn fine-grained discriminative features, then optimizes a concept-level head to align semantics without altering the learned representation. This concentrates the surrogate decision boundaries on $\mathcal{S}_\cap$, increasing the likelihood that AEs transfer to both F^P and F^T despite differences in label semantic granularity.

Multi-Level Objective for Transfer Attack. Given a fine-tuned surrogate model F^S with a concept-level classifier head $\tilde{F}^S$, we craft an AE $x^* = x + \delta$ from a clean surrogate sample $x \in D^S$ with ground-truth label y^S, semantic concept $\phi_S(y^S)$, and aligned labels y^P and y^T under F^P and F^T, respectively, subject to $||\delta||p \le \epsilon$ (see Fig. 3). The perturbation δ is obtained by maximizing the joint attack loss in (14), which induces misclassification on F^S while promoting transferability to the black-box victim models. The coefficients λ_{tas}, λ_{con}, and $\lambda_{hid} \ge 0$ balance the contributions of the respective loss terms.

$$\max_{||\delta||_p \le \epsilon} \lambda_{tas}\mathcal{L}_{tas} + \lambda_{con}\mathcal{L}_{con} + \lambda_{hid}\mathcal{L}_{hid}. \tag{14}$$

(a) Surrogate Task Loss. The task-level loss $\mathcal{L}_{tas}$, defined in (15), is a standard adversarial loss (e.g., cross-entropy) applied to the output of F^S. Maximizing $\mathcal{L}_{tas}$ drives the AE bypass the surrogate-task specific decision boundary of F^S.

$$\mathcal{L}_{tas} = \mathcal{L}(F^S(x+\delta), y^S), \tag{15}$$

(b) Semantic Concept Loss. The concept-level loss $\mathcal{L}_{con}$, defined in (16), operates on the semantic concept prediction head $\tilde{F}^S$, which maps the output of F^S to overlapped semantic concepts $\phi_S(y^S)$. Unlike $\mathcal{L}_{tas}$, which is tied to the surrogate label space, $\mathcal{L}_{con}$ encourages perturbations that disrupt shared semantic concepts across surrogate and victim models, thereby improving transferability across different label spaces.

$$\mathcal{L}_{con} = \mathcal{L}(\tilde{F}^S(F^S(x+\delta)), \phi_S(y^S)) \tag{16}$$

(c) Hidden Representation Loss. To mitigate surrogate-task overfitting, we introduce a hidden representation loss $\mathcal{L}_{hid}$ that maximizes feature-level discrepancies between x and x^* at an intermediate layer of F^S. Such hidden representations, particularly in earlier layers, capture low-level patterns that are more architecture-agnostic and dataset-invariant than logits. Formally, given the hidden layers $\{h_l\}_{l=1}^L$ of F^S, we select a layer h_l (with default $l = \lfloor L/3 \rfloor$) and define $\mathcal{L}_{hid}$ as in (17).

$$\begin{aligned}\mathcal{L}_{hid} &= (h_l(x+\delta) - h_l(x)) \cdot ({}^{d}/_{||d||_2}) \\ &= ||h_l(x+\delta) - h_l(x)||_2 \cos(h_l(x+\delta) - h_l(x), d)\end{aligned} \tag{17}$$

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

Algorithm 1 Hierarchical Fine-tuning of Surrogate Model

Input: Surrogate dataset $D^S = \{(x^S, y^S)\}$ aligned with $\mathcal{S}_\cap$; Training split $D^S_{\text{train}} \subseteq D^S$; Semantic mapping ϕ_S; Maximum number of epochs T; learning rate η.
Output: surrogate model F^S and concept head $\tilde{F}^S$

Stage I: Task-level Fine-tuning
Initialize $F^S(\cdot;\theta)$ from a public pretrained model
for epoch $= 1, \ldots, T$ **do**
  for each batch $\mathcal{B} = \{(x_i^S, y_i^S)\}_{i=1}^{|\mathcal{B}|}$ in D^S_{train} **do**
    Calculate cross-entropy loss: $L \leftarrow \frac{1}{|\mathcal{B}|} \sum_{(x_i^S, y_i^S)\in\mathcal{B}} \mathcal{L}(F^S(x_i^S;\theta), y_i^S)$
    Update parameters θ in F^S: $\theta \leftarrow \theta - \eta \nabla_\theta L$
  end for
end for
Stage II: Concept-level Fine-tuning
Relabel D^S_{train} with semantic labels: $D^S_{\text{train}} \leftarrow \{(x^S, \phi_S(y^S)) : (x^S, y^S) \in D^S_{\text{train}}\}$
Freeze F^S parameters θ
Append semantic concept head $\tilde{F}^S(\cdot;\psi)$
for epoch $= 1, \ldots, T$ **do**
  for each batch $\mathcal{B} = \{(x_i^S, \phi_S(y_i^S))\}_{i=1}^{|\mathcal{B}|}$ in D^S_{train} **do**
    Calculate cross-entropy loss: $L \leftarrow \frac{1}{|\mathcal{B}|} \sum_{(x_i^S, \phi_S(y_i^S))\in\mathcal{B}} \mathcal{L}(\tilde{F}^S(F^S(x_i^S);\psi), \phi_S(y_i^S))$
    Update parameters ψ in $\tilde{F}^S$: $\psi \leftarrow \psi - \eta \nabla_\psi L$
  end for
end for
return $F^S, \tilde{F}^S$

where $d = \nabla_{h_l} \mathcal{L}(F^S(x), y^S)\big|_{h_l = h_l(x)}$ is the gradient of the classification loss $\mathcal{L}$ w.r.t h_l. $\mathcal{L}_{hid}$ drives the hidden representation of the perturbed input $h_l(x+\delta)$ away from the clean one along the gradient direction induced by $h_l(x)$.

By jointly optimizing $\mathcal{L}_{tas}$, $\mathcal{L}_{con}$, and $\mathcal{L}_{hid}$, the resulting AEs gain greater flexibility for simultaneous transfer to diverse victims. $\mathcal{L}_{tas}$ induces fine-grained semantic shifts, favoring transfer to models with fine-grained label spaces and high task similarity to the surrogate, while $\mathcal{L}_{con}$ disrupts coarse-grained semantic attributes, facilitating transfer to models with coarser label spaces. Together, these complementary objectives enable AEs to transfer effectively to black-box models with diverse architectures and label granularities in downstream adaptation.

B. Attack Workflow

Given the conceptually aligned surrogate dataset D^S and the hierarchically fine-tuned surrogate model F^S with concept head $\tilde{F}^S$, `TriXfer` proceeds in three stages (Algorithm 2).

- *Stage I: Adversarial Example Generation.* For each surrogate test sample $(x^S, y^S) \in D^S_{\text{test}}$ augmented with its concept label $\phi_S(y^S)$, a perturbation δ is crafted on F^S and $\tilde{F}^S$ by optimizing the joint objective in (14).
- *Stage II: Transfer Attack on the Pretrained ViT.* The generated AEs are transferred to the black-box F^P (success rate ASR^P). AEs that induce misclassification on F^P are retained as the pretrained-effective set for assessing black-box vulnerability propagation to downstream models.
- *Stage III: Transfer Attack on the Downstream Model.* Applying the pretrained-effective AEs to F^T with aligned labels y^T yields the conditional success rate $\text{ASR}^{\text{con}T}$, revealing inherited vulnerability. The full AE set (from Stage I) applied to F^T yields the overall downstream rate ASR^T.

Algorithm 2 TriXfer Attack Under Downstream Adaptation

Input: Surrogate dataset $D^S = \{(x^S, y^S)\}$ aligned with $\mathcal{S}_\cap$; Test split $D^S_{\text{test}} \subseteq D^S$; Surrogate model F^S with concept-level classifier head $\tilde{F}^S$; Semantic mapping functions ϕ_S, ϕ_P, ϕ_T; Black-box access to pretrained model F^P and downstream model F^T.
Output: Downstream attack success rate ASR^T

Stage 1: Adversarial Example Crafting
Initialize adversarial examples set $\mathcal{A} \leftarrow \emptyset$
Relabel D^S_{test} with semantic labels: $D^S_{\text{test}} \leftarrow \{(x^S, y^S, \phi_S(y^S)) : (x^S, y^S) \in D^S_{\text{test}}\}$
for each $(x^S, y^S, \phi_S(y^S)) \in D^S_{\text{test}}$ **do**
  Craft adversarial example $x^* = x^S + \delta$ using (14)
  Add $(x^*, x^S, y^S, \phi_S(y^S))$ to $\mathcal{A}$
end for
Stage 2: Transfer Attack Pretrained ViT Model
Assign semantically aligned labels from $\mathcal{Y}^P$ to samples in $\mathcal{A}$: $\mathcal{A} \leftarrow \{(x^*, x^S, y^S, y^P) : (x^*, x^S, y^S, \phi_S(y^S)) \in \mathcal{A},\ \phi_P(y^P) = \phi_S(y^S)\}$
Initialize adversarial examples set $\mathcal{A}^* \leftarrow \emptyset$
for each $(x^*, x^S, y^S, y^P) \in \mathcal{A}$ **do**
  if $F^P(x^*) \neq y^P$ **then**
    Add (x^*, x^S, y^S, y^P) to $\mathcal{A}^*$
  end if
end for
Compute ASR of F^P: $\text{ASR}^P \leftarrow \frac{|\mathcal{A}^*|}{|\mathcal{A}|}$
Stage 3: Transfer Attack Downstream Model
Assign semantically aligned labels from $\mathcal{Y}^T$ to samples in $\mathcal{A}^*$: $\mathcal{A}^* \leftarrow \{(x^*, x^S, y^S, y^T) : (x^*, x^S, y^S, y^P) \in \mathcal{A}^*,\ \phi_T(y^T) = \phi_S(y^S)\}$
Initialize downstream attack success counter $c_T \leftarrow 0$
for each $(x^*, x^S, y^S, y^T) \in \mathcal{A}^*$ **do**
  Query downstream model prediction $F^T(x^*)$
  if $F^T(x^*) \neq y^T$ **then**
    $c_T \leftarrow c_T + 1$
  end if
end for
Compute conditional ASR of F^T: $\text{ASR}^{\text{con}T} \leftarrow \frac{c_T}{|\mathcal{A}^*|}$
return ASR^T

V. Experiments

We first describe the experimental setup (§V-A), LLM-based label alignment (§V-B), and model generation (§V-C). Then, we present results addressing RQ1 (§V-D) and RQ2 (§V-E), and compare `TriXfer` with attack baselines (§V-F). Subsequently, we provide class-wise attack analysis (§V-G), and LLM-alignment robustness study (§V-H). Finally, we present ablation studies (§V-I) and concept-level alignment analysis (§V-J). Code is available at https://github.com/TriXfer.

Experiments are implemented in PyTorch 2.7.1 [50] (CUDA 11.8) on an NVIDIA RTX 3090 GPU, using pretrained models from timm (v1.0.22) [51] and a fixed random seed of 42.

A. Experimental Setup

1) Models

Pretrained Models. We evaluate diverse ViT-family backbones spanning architectures and pretraining paradigms: global-self-attention Transformers (ViT-B/16 [3], CaiT-B [24], DeiT III-B [22], DINOv2 ViT-S [23], CLIP ViT-B/16, and CLIP ViT-L/14) and hybrid architectures with convolutional inductive biases (Swin-B/P4W7 [29] and TinyViT [25]). These span supervised (ViT-B, CaiT-B, DeiT III-B, Swin, TinyViT),

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

self-supervised (DINOv2), and vision–language contrastive (CLIP ViT-B, CLIP ViT-L) pretraining. To study robustness-enhanced pretraining, we additionally evaluate adversarially pretrained AT-ViT-B/16 and AT-Swin-B/P4W7 [30], keeping architectures identical to their standard counterparts.

Downstream Models. We adapt each of the above 10 pretrained ViTs to downstream classification under all six adaptation mechanisms: full fine-tuning (FullFT), linear probing (LinearProb), adapter tuning (AdapterT) [26], LoRA [27], prompt tuning (PromptT), and prefix tuning (PrefixT) [28].

Surrogate Models. We use surrogate architectures that differ from the victim backbones, including modern CNNs (ConvNeXt V2 [7], InceptionNeXt [31]) and lightweight ViTs (MobileViT [32], EfficientViT [33]), each hierarchically fine-tuned on the surrogate dataset following Algorithm 1.

2) Datasets

Pretraining Dataset. All pretrained models use weights trained on ImageNet-1K (a.k.a ILSVRC2012, 1.28M 224×224 images) across 1,000 classes [52]. This dataset provides broad semantic coverage and serves as the pretraining source for both standard and adversarially pretrained ViTs in our experiments.

Downstream Datasets. We evaluate on CIFAR-10 and CIFAR-100 [40] (60K 32×32 images, 10 coarse vs. 100 fine-grained classes), and ImageNette-10 [41] (300×500 images, 10 classes). All images are resized to 224×224 by downsampling or upsampling to match the pretrained ViTs adaptation.

Surrogate Datasets. Following our adversary model (§III), we select surrogate datasets by the victim's disclosed semantic scope, not its specific label names. We use surrogate domains whose label spaces align with the overlapping concepts between the pretraining and downstream datasets: STL-10 [38], CINIC-10 [39], Oxford-IIIT Pets-37 [34], Stanford Cars-196 [35], FGVC-Aircraft-100 [36], and ImageWoof-10 [37], enabling a *cross-dataset* and *cross-label-space* transfer setting.

These surrogates exhibit two types of label semantic overlap with respect to the victims:

- *Hierarchical overlap*. Fine-grained subclasses (pet/dog breeds, car models, aircraft variants in OxfordPets, ImageWoof, StanfordCars, FGVCAircraft) map to target superclasses (e.g., *dog*, *cat*, *automobile*, *airplane* in CIFAR-10);
- *Non-hierarchical overlap*. Surrogate and target labels share comparable semantic abstraction (e.g., STL-10 and CINIC $\rightarrow$ CIFAR-10), enabling transfer attacks across datasets with different distributions but consistent label granularity.

3) Training Details

All models are trained for 50 epochs with AdamW and cosine annealing ($T_{\max} = 50$). Batch size is 128 (64 for PromptT and PrefixT, due to memory). Weight decay is 1×10^{-4}, except 1×10^{-2} for FullFT to reduce overfitting. Learning rates are tuned per method and dataset: FullFT 1×10^{-4} and LinearProb 1×10^{-2} on both CIFAR-10/100; AdapterT and LoRA 1×10^{-3} on CIFAR-10 and 1–3×10^{-3} on CIFAR-100; PromptT $1/1.5\times10^{-3}$ and PrefixT $1/2\times10^{-3}$ on CIFAR-10/100. LoRA uses rank 8, scaling 16, and dropout 0.1, injected into `qkv`, `proj`, and MLP layers; AdapterT uses a bottleneck dimension of 64 on the same modules. PromptT and PrefixT use 10 tokens each, with extra projection layers for windowed-attention backbones (Swin, TinyViT).

4) Transfer Attack Baselines

We compare `TriXfer` against leading transfer attacks (SP [9], cross-domain UAP [42], and XTransfer [43]) under a single surrogate model, all generating AEs on the clean surrogate domain. Each is combined with five perturbation operators: FGSM [44], PGD (aka, IFGSM) [45], MIFGSM (aka, MPGD) [46], DIFGSM [47], and TIFGSM [48].

5) Attack Settings

All attacks are untargeted under L_∞ with $\epsilon = 16/255$; iterative attacks use 20 steps of $\alpha = 4/255$. MIFGSM uses momentum 0.9; DIFGSM uses diversity probability 0.5 and resize rate 0.9; TIFGSM applies Gaussian gradient smoothing. For `TriXfer`, we optimize the joint objective in (14) with default weights $\lambda_{tas} = \lambda_{con} = 0.25$, $\lambda_{hid} = 0.5$ (hierarchical) and $\lambda_{tas} = 0.25$, $\lambda_{hid} = 0.75$ (non-hierarchical). $\mathcal{L}_{hid}$ is computed at one-third depth, $l = \lfloor L/3 \rfloor$. These default weights are justified by a systematic ablation (§V-I3).

6) Evaluation Metrics

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be the surrogate test set. AEs $x_i^* = x_i + \delta$ are crafted from clean samples x_i on the surrogate model with concept-aligned labels y_i^P, y_i^T, and transferred to the ViT foundation and downstream models. We evaluate the success rate of transfer attacks and also report victim misclassification on clean surrogate samples to confirm non-trivial transfer.

Standard metrics. The attack success rate (ASR) defined in (18) measures the proportion of AEs that cause misclassification in the victim model, where $\mathbb{I}(\cdot)$ is the indicator function.

$$\mathrm{ASR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(F(x_i^*) \neq y_i) \tag{18}$$

The clean accuracy (ACC) of a model F is defined as (19).

$$\mathrm{ACC} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(F(x_i) = y_i) \tag{19}$$

We denote the evaluation results on F^P as ASR^P and ACC^P, and those on F^T as ASR^T and ACC^T, respectively.

Conditional metrics. To examine vulnerability inheritance, we further evaluate F^T on two conditional subsets of $\mathcal{D}$: (i) AEs that successfully attack F^P, $\mathcal{D}^{\mathrm{ConAdv}} = \{(x_i^*, y_i)) \in \mathcal{D} \mid F^P(x_i^*) \neq y_i^P\}$; and (ii) clean samples misclassified by F^P, $\mathcal{D}^{\mathrm{ConCle}} = \{(x_i, y_i) \in \mathcal{D} \mid F^P(x_i) \neq y_i^P\}$. The conditional ASR (**ConASR**) on F^T is defined as (20).

$$\mathrm{ASR}^{\mathrm{con}T} = \frac{1}{|\mathcal{D}^{\mathrm{ConAdv}}|}\sum_{(x_i,y_i)\in\mathcal{D}^{\mathrm{ConAdv}}}\mathbb{I}\big(F^T(x_i^*) \neq y_i\big) \tag{20}$$

The conditional ACC (ConACC) on F^T is defined as (21).

$$\mathrm{ACC}^{\mathrm{con}T} = \frac{1}{|\mathcal{D}^{\mathrm{ConCle}}|}\sum_{(x_i,y_i)\in\mathcal{D}^{\mathrm{ConCle}}}\mathbb{I}\big(F^T(x_i) = y_i\big) \tag{21}$$

B. LLM-based Concept Alignment across Label Spaces

We perform LLM-based concept alignment, using LLMs as external semantic tools to map labels from different label spaces into a shared concept space $\mathcal{S}$ and expose their concept-level overlap. Across $\mathcal{Y}^P$ (ImageNet), $\mathcal{Y}^T$ (CIFAR-10/100, ImageNette), and $\mathcal{Y}^S$ (OxfordPets, StanfordCars, FGVCAircraft,

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

TABLE II
MAPPING SURROGATE DATASET LABELS TO IMAGENET AND CIFAR-10 LABEL SPACES

CIFAR-10	ImageNet-1K (Representative)	Semantic Concept	STL-10	CINIC	Oxford-IIIT Pets	Combined-CAW Stanford Cars	Combined-CAW FGVC-Aircraft	ImageWoof
automobile	car, sports car, sedan, SUV	Automobile / Car	✓	✓	✗	✓	✗	✗
bird	eagle, sparrow, parrot, finch	Bird	✓	✓	✗	✗	✗	✗
cat	tabby cat, Persian cat, Siamese cat	Cat	✓	✓	✓	✗	✗	✗
dog	golden retriever, husky, terrier, poodle	Dog	✓	✓	✓	✗	✗	✓
deer	elk, antelope, deer	Deer / Ungulate	✓	✓	✗	✗	✗	✗
frog	tree frog, bullfrog	Frog / Amphibian	✗	✓	✗	✗	✗	✗
horse	horse, zebra	Horse / Equine	✓	✓	✗	✗	✗	✗
ship	container ship, ferry, boat	Ship / Boat	✓	✓	✗	✗	✗	✗
truck	pickup truck, lorry	Truck	✓	✓	✗	✗	✗	✗

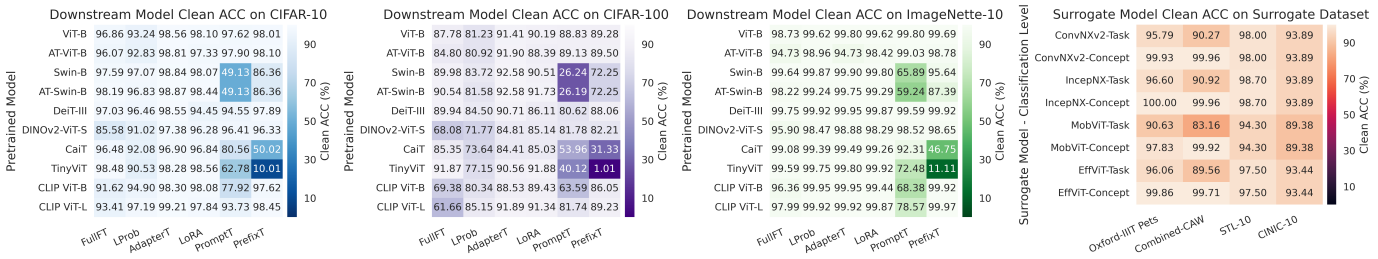


Fig. 4. Clean performance of downstream and surrogate models. Task and concept denote the performance of the label and concept classifiers, respectively.

ImageWoof, STL, CINIC), labels with overlapping semantics but different granularities map to the same concept in $\mathcal{S}$.

Label-to-concept mapping. For each label in $\mathcal{Y}^P$, $\mathcal{Y}^T$, and $\mathcal{Y}^S$, we query an LLM to obtain its concept, inducing the label-to-concept mappings $\phi_P : \mathcal{Y}^P \to \mathcal{S}$, $\phi_T : \mathcal{Y}^T \to \mathcal{S}$, and $\phi_S : \mathcal{Y}^S \to \mathcal{S}$. The shared concepts are the intersection of the mapped concepts across $\mathcal{Y}^P$, $\mathcal{Y}^T$, and $\mathcal{Y}^S$ (Table II gives an example with CIFAR-10 as the downstream $\mathcal{Y}^T$).

Cross-LLM validation. To reduce bias and hallucination from any single LLM, we adopt a cross-validation strategy using three independent LLMs (Claude, Gemini, and ChatGPT) and retain the label-to-concept mappings consistent across them. When multiple labels map to the same concept, we retain all of them, yielding a many-to-one label-to-concept mapping.

Attack Success Criterion. Attack success depends on the alignment between the surrogate and victim label spaces. For each AE: (i) under hard label-level alignment, success means the victim prediction differs from the true hard label; (ii) when only concept-level alignment exists, success means the prediction falls outside the label set of the aligned true concept.

C. Downstream and Surrogate Models Generation

We report the clean performance of downstream and surrogate models to verify proper training before attack evaluation.

1) Downstream Adaptation Performance

The clean Top-1 accuracy of downstream models adapted from different pretrained ViTs is shown in Fig. 4. Across CIFAR-10, CIFAR-100, and ImageNette, most configurations converge strongly (many exceeding 95% on CIFAR-10), providing a reliable basis for attack evaluation.

AdapterT and LoRA reach strong clean accuracy, often surpassing FullFT across backbones and datasets (e.g., AdapterT vs. FullFT on ViT-B/CIFAR-10, 98.56% vs. 96.86%). PromptT and PrefixT, however, underfit on some architectures: on CIFAR-10, PromptT on Swin drops to 49.13% and PrefixT on TinyViT to 10.01%, though these backbones exceed 98% with AdapterT or LoRA. This reflects structural incompatibility rather than insufficient TL: window attention limits prompt propagation in Swin and TinyViT, and CaiT's class-attention weakens prompt/prefix influence. We therefore focus subsequent attacks on reliably converged models and interpret underfitted cases cautiously. Adversarially pretrained backbones (AT-ViT-B, AT-Swin-B) retain clean accuracy close to their standard counterparts. This confirms that *robustness-enhanced pretraining* does not significantly degrade generalization, allowing later attack effectiveness differences to be attributed to robustness rather than clean accuracy loss. Both CLIP backbones achieve strong clean accuracy. Their vision-language pretraining objective makes them a meaningful test of attack generality beyond only vision modality-based ViTs.

2) Hierarchical Fine-tuning of Surrogate Models

We report surrogate clean accuracy across different models and datasets at the task level (original surrogate labels) and concept level (mapped concept). All four surrogates (ConvNeXt, IncepNeXt, MobViT, EffViT) trained well (see Fig. 4).

For hierarchical overlap, OxfordPets reaches 94.77% task and 99.41% concept accuracy (avg). On the more challenging **Combined-CAW** dataset (a combined dataset of StanfordCars, FGVCAircraft, and ImageWoof), task accuracy drops to 88.48% (due to intra-class diversity) yet concept accuracy stays near-perfect (99.89%), showing that concept-level fine-tuning abstracts fine-grained predictions into stable concepts.

For non-hierarchical overlap, task and concept accuracies coincide (concepts equal dataset labels), averaging 97.13% on STL-10 and 92.65% on CINIC. Overall, hierarchical fine-tuning yields reliable surrogates with both fine-grained and coarse-grained feature discrimination.

D. Transferability in a Dual Black-Box Setting

To address *'RQ1:Do black-box downstream classifiers remain vulnerable to transferable AEs that can successfully mislead a black-box pretrained ViT?'*, we report both ASR and

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

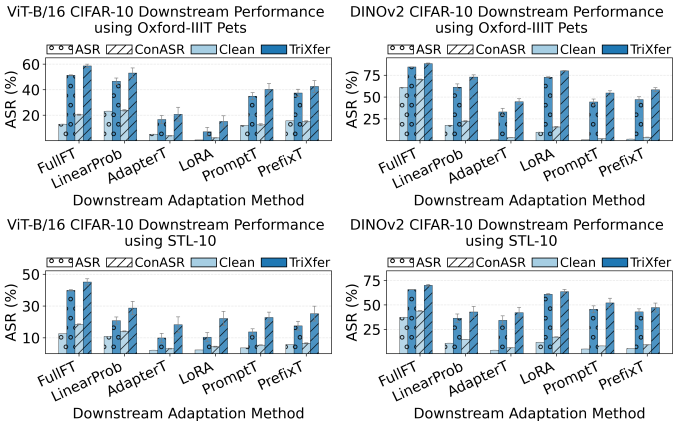


Fig. 5. Attack downstream models under various adaptations. Bars show the average over four surrogate models, and error bars indicate variability.

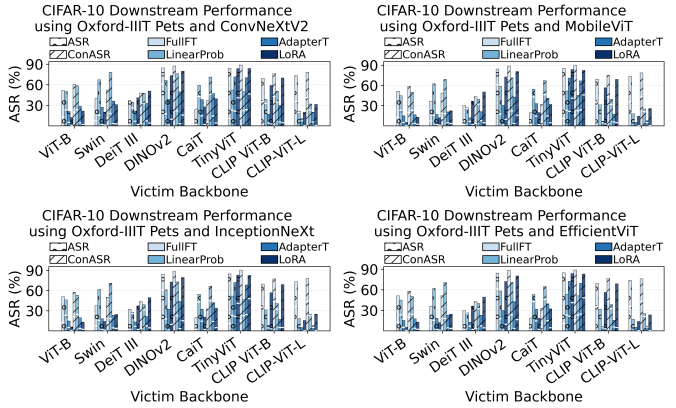


Fig. 6. Attack downstream models with various downstream architectures. White line shows downstream misclassification rate on clean surrogate data.

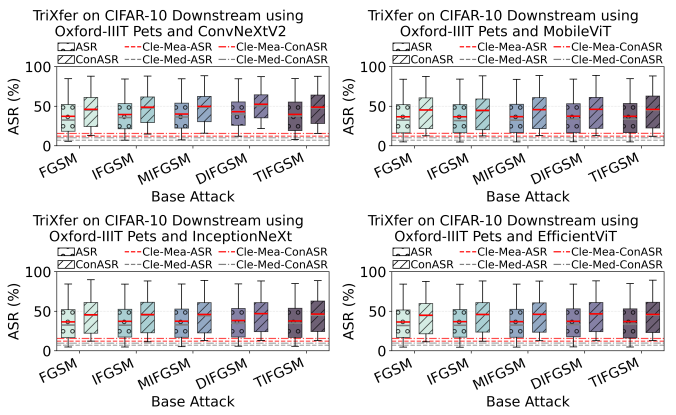


Fig. 7. Attack downstream models with various basic perturbation operators. In the box plot, red line shows the mean, and gray line shows the median.

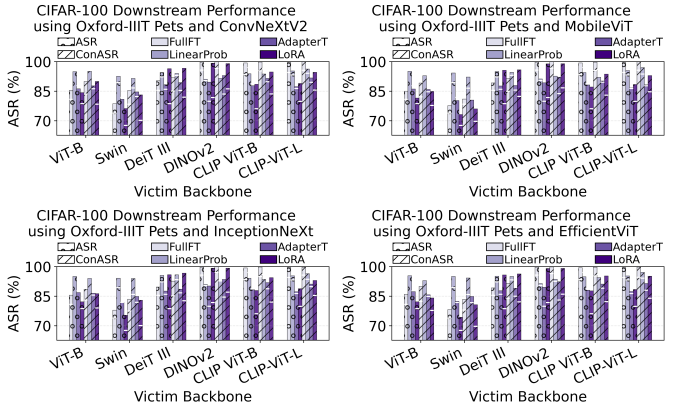


Fig. 8. Attack downstream models on various downstream datasets. White line shows downstream misclassification rate on clean surrogate data.

ConASR of `TriXfer` on downstream models. ASR measures the overall downstream misclassification rate, while ConASR conditions on AEs that already succeed on F^P.

1) Attack on Various Downstream Adaptation Methods

Setup. Fixing the victim backbone to ViT-B and DINOv2, each adapted to CIFAR-10 under six methods (FullFT, LinearProb, AdapterT, LoRA, PromptT, PrefixT), we use OxfordPets and STL-10 as surrogate datasets (hierarchical overlap) with four surrogate architectures (ConvNeXt, IncepNeXt, MobViT, EffViT) and DIFGSM-based `TriXfer` AEs (Fig. 5).

Result. Clean surrogate data yield low downstream misclassification across all adaptations (ViT-B 0.6%–23.2% on Oxford, 2.2%–12.6% on STL; DINOv2 1.1%–61.0% and 3.6%–37.3%), confirming a non-trivial triple-cross transfer setting. `TriXfer` AEs sharply increase vulnerability under every adaptation and surrogate: ASR rises to 7.1%–51.3% and ConASR to 15.2%–58.7% on ViT-B, and higher on DINOv2 (33.0%–84.5% and 44.7%–88.3%), consistently across Oxford and STL. No adaptation prevents transfer, and the consistently higher ConASR shows that pretrained-backbone vulnerabilities largely persist through various adaptation methods.

2) Attack on Various Downstream Model Architectures

Setup. Fixing the perturbation operator as previously, we evaluate `TriXfer` on downstream models adapted from diverse ViT-family backbones: ViT-B, Swin, DeiT III, DINOv2, CaiT, TinyViT, CLIP ViT-B, and CLIP ViT-L (Fig. 6).

Result. Clean surrogate data cause low misclassification for most architecture–adaptation combinations (typically below 25%), apart from the structural underfitting of certain backbones (Swin, CaiT, TinyViT under specific PromptT/PrefixT); architectural variation thus does not induce systematic clean-data errors. `TriXfer` AEs transfer strongly across all architectures (ASR 10.8%–84.7%, ConASR frequently $>$ 50%): ASR varies by architecture, but every model is markedly vulnerable, and the higher ConASR confirms that AEs effective on the pretrained ViTs remain effective after adaptation.

3) Attack with Various Basic Perturbation Operators

Setup. We vary `TriXfer`'s basic perturbation operator over FGSM, IFGSM, MIFGSM, DIFGSM, and TIFGSM (Fig. 7), known for strong adversarial transferability. Downstream ASR and ConASR are pooled across victim backbones (ViT-B, Swin, DeiT III-B, DINO, CLIPs) and adaptations (FullFT, LinearProb, AdapterT, LoRA), one box per operator.

Result. Across all perturbation operators, pooled downstream ASR far exceeds the clean baseline (median ASR 7%, dashed lines): median ASR is typically 33%–40%, with higher ConASR (45%–50%), indicating uniformly elevated attack success rather than a few strong outliers. `TriXfer` thus demonstrates robust transferability across diverse perturbation operators, downstream and surrogate architectures, and adaptation strategies, without relying on any specific operator, though MIFGSM and DIFGSM give the most consistent transfer.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

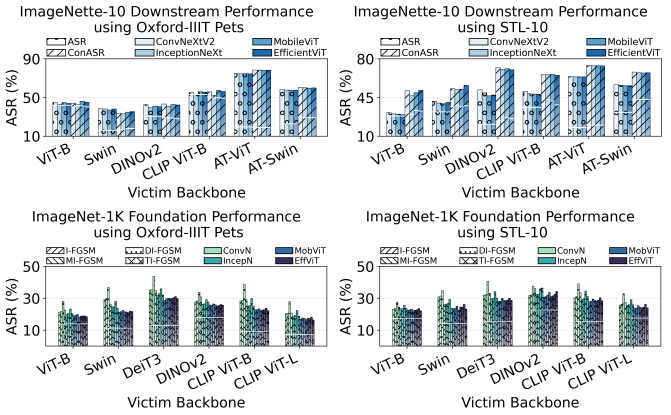


Fig. 9. Attack victims on high-resolution victim domains: ImageNette and ImageNet. White line shows misclassification rate on clean surrogate data.

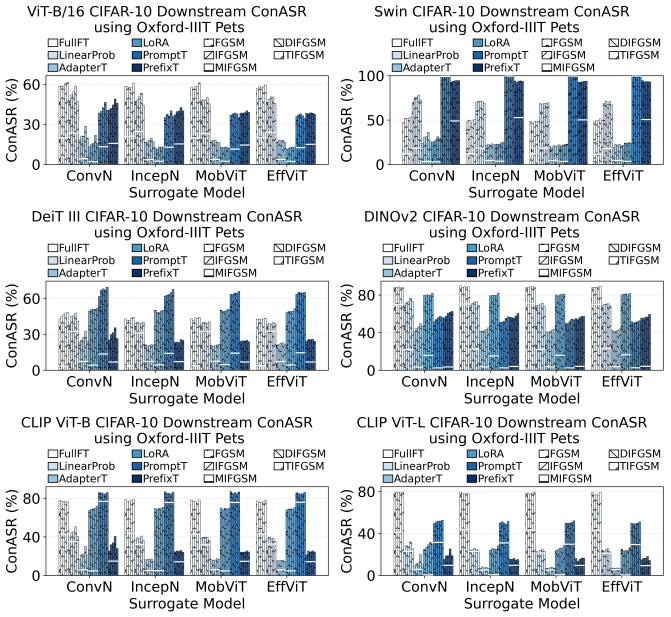


Fig. 10. Attack downstream models with various surrogate architectures. White line shows downstream misclassification rate on clean surrogate data.

4) Attack on Various Downstream Datasets

Setup. We further test `TriXfer` beyond downstream CIFAR-10 along two axes: (i) a harder downstream task, CIFAR-100 (Fig. 8); and (ii) higher-resolution downstream ImageNette-10 (Fig. 9). We also evaluate the high-resolution, high-complexity ImageNet-1K foundation domain.

Result. On CIFAR-100, clean misclassification is already high (often over 68%) due to task difficulty, and `TriXfer` pushes ASR to 76%–100% and ConASR to 83%–100% (largest gains where clean misclassification leaves room, e.g., FullFT, LoRA). On ImageNette-10, `TriXfer` transfers strongly across all backbones and surrogates (ASR up to 80%, higher ConASR), confirming effectiveness across downstream datasets of differing complexity. On ImageNet-1K, it fools the pretrained ViTs across all backbones and operators (ASR up to 45%) under both surrogates, showing that `TriXfer` remains effective at higher resolution and harder tasks.

5) Attack with Various Surrogate Architectures

Setup. We fix the surrogate and downstream dataset and vary only the surrogate architecture (Fig. 10), generating AEs with ConvNeXtV2, IncepNeXt, MobViT, and EffViT; downstream models are adapted from multiple ViTs (e.g., ViT-B/16, Swin, DeiT III, DINOv2, CLIP ViT-B, CLIP ViT-L, etc).

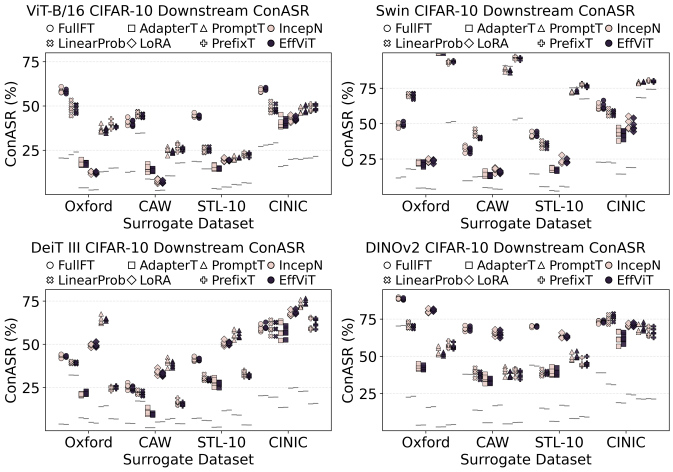


Fig. 11. Attack downstream models with various surrogate datasets. Gray line shows downstream misclassification rate on clean surrogate data. ConASR values in the same column originate from different perturbation operators.

Result. `TriXfer` achieves high downstream ConASR across all four surrogate architectures (44%–90%), well above the clean baseline (14%). While variation exists across surrogates, no surrogate architecture consistently dominates: both CNN-based and compact lightweight surrogates (MobViT, EffViT) enable strong transfer, indicating that `TriXfer` does not depend on surrogate architecture or capacity.

6) Attack with Various Surrogate Datasets

Setup. We vary the surrogate dataset with all other settings fixed (Fig. 11): OxfordPets, CAW, STL-10, and CINIC, using IncepNeXt and EffViT surrogates and CIFAR-10 victims adapted from ViT-B, Swin, DeiT III-B, DINO, and CLIPs.

Result. `TriXfer` achieves high downstream ConASR across all surrogate datasets, well above clean baselines. (i) Hierarchical-overlap surrogates transfer well (Oxford-Pets 12.8%–88.5%, Combined-CAW 6.9%–68.5%); (ii) non-hierarchical surrogates, with statistics closer to CIFAR-10, give more concentrated transfer (STL-10 15.1%–69.7%, comparable to hierarchical surrogates; CINIC highest and tightest, 42.8%–78.6%). Overall, `TriXfer` is robust to substantial surrogate dataset variation, showing that strong transfer needs neither strict label alignment nor identical data distributions.

Takeaway of RQ1:

- *Downstream classifiers are highly vulnerable to transferable AEs crafted against pretrained ViT models, even when both the pretrained and downstream models are black boxes.*
- *Across extensive evaluations, `TriXfer` consistently achieves high downstream ASR and ConASR under substantial variations in ViT backbones, adaptation strategies, and datasets.*
- *This vulnerability persists across perturbation operators, surrogate models, and surrogate datasets, including challenging lightweight surrogates and hierarchically aligned label spaces.*
- *Overall, these results indicate that the black-box adversarial vulnerabilities of ViT foundation models pretrained on purely visual or vision-language data are widely inherited by downstream models.*

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

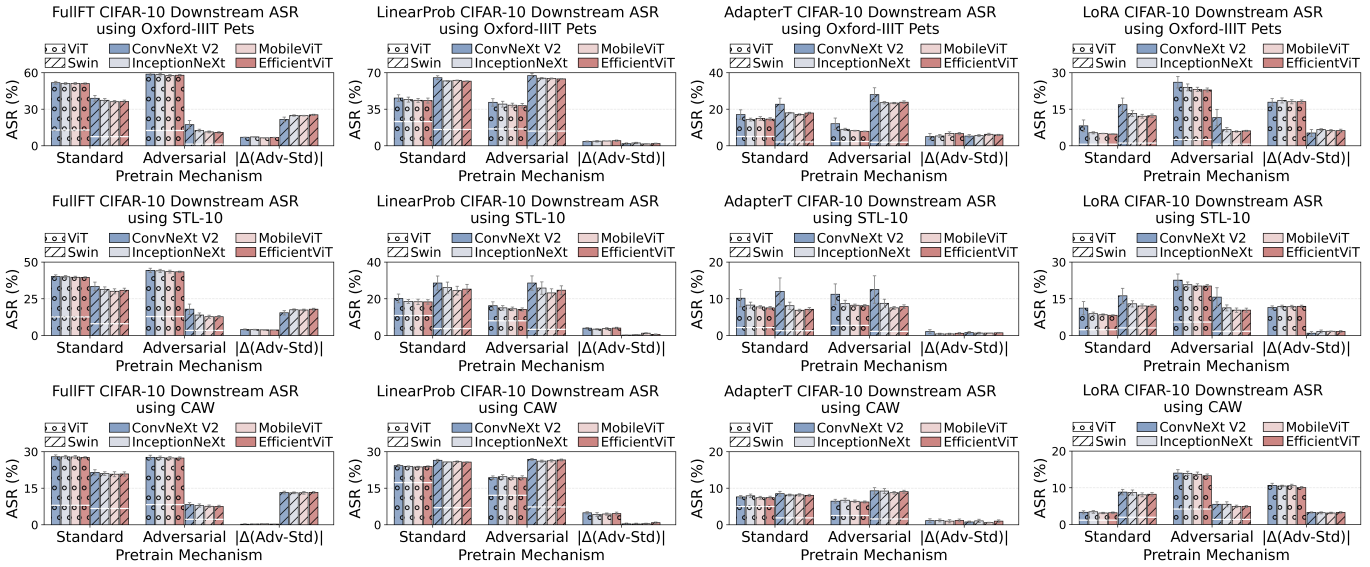


Fig. 12. Impact of pretraining mechanisms on downstream robustness. Bars show average downstream ASR over five basic perturbation operators and error bars indicate variability. White line shows downstream misclassification rate on clean surrogate data.

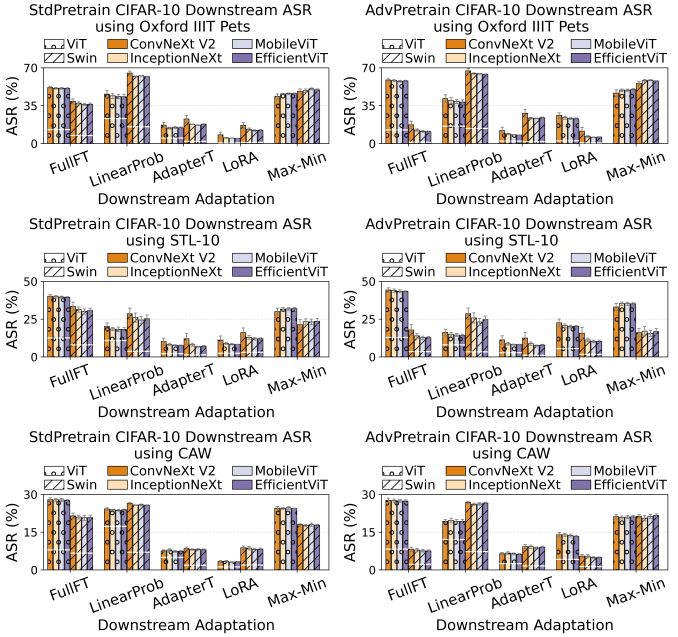


Fig. 13. Impact of adaptation methods on downstream robustness. Bars show average ASR over five basic perturbation operators; error bars indicate variability. White line shows misclassification rate on clean surrogate data.

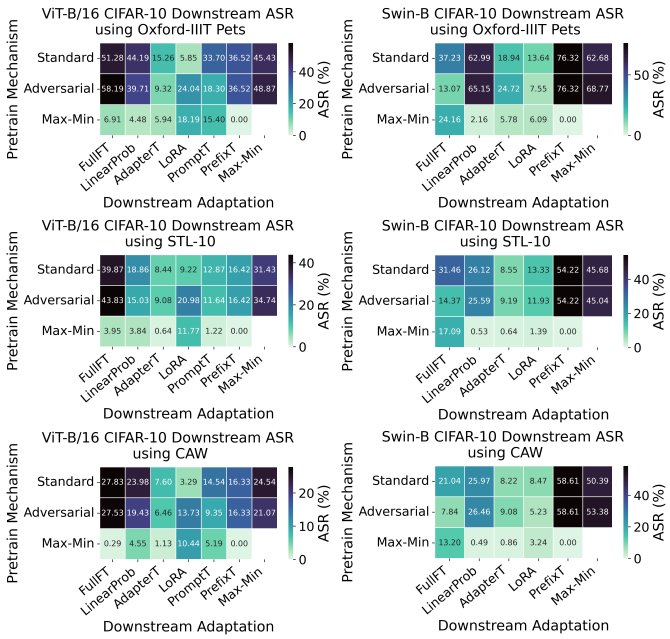


Fig. 14. Interaction effects of pretraining mechanism and downstream adaptation strategy on downstream model robustness.

E. Factors Influencing Downstream Robustness

To address *'RQ2: Is downstream robustness primarily determined by the pretraining mechanism or the downstream adaptation strategy?'*, we analyze downstream robustness by disentangling the effects of pretraining mechanisms and downstream adaptation methods.

1) Impact of Pretraining Mechanism

Setup. We compare standard pretraining (StdPretrain) and adversarial pretraining (AdvPretrain) of the backbone while keeping the downstream adaptation strategy fixed (see Fig. 12). Transferred AEs are generated using `TriXfer`.

Result. Switching from StdPretrain to AdvPretrain causes observable but limited ASR changes on both backbones. For ViT with LinearProb or AdapterT, and Swin with FullFT or LoRA, AdvPretrain gives slightly lower or comparable ASR, suggesting partial carry-over of pretraining robustness. However, the effect is inconsistent: under LoRA, AdvPretrain can even raise ASR, and the sign of the difference varies by backbone. Thus the pretraining mechanism influences downstream robustness only weakly and unstably, and does not dominate vulnerability when the adaptation is fixed.

2) Impact of Downstream Adaptation Method

Setup. We then fix the pretraining mechanism (StdPretrain or AdvPretrain) and vary the adaptation method (FullFT, LinearProb, AdapterT, LoRA) on ViT-B and Swin-B (Fig. 13).

Result. Adaptation methods cause much larger ASR variation than pretraining. FullFT and LinearProb give the highest ASR regardless of pretraining, while AdapterT and LoRA consistently lower it by restricting updates to the pretrained rep-

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

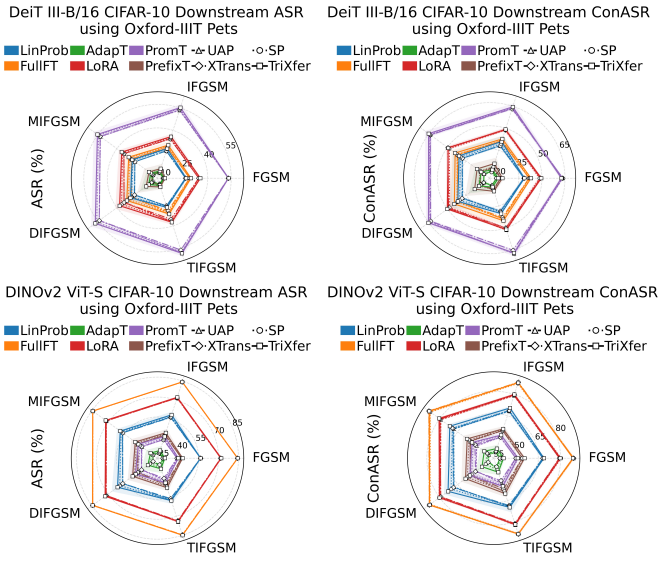


Fig. 15. Comparison with transfer attack baselines (Oxford-IIIT Pets → CIFAR-10). Shading shows variability across four surrogate architectures.

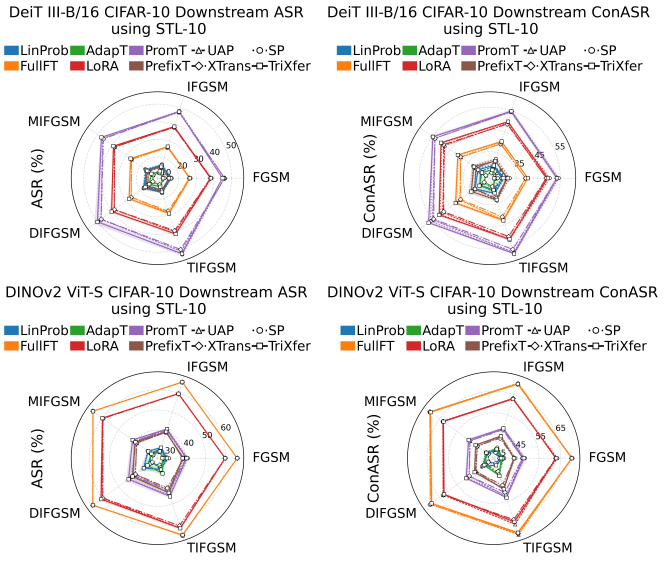


Fig. 16. Comparison with transfer attack baselines (STL-10 → CIFAR-10). Shading shows variability across four surrogate architectures.

resentations. The ASR range (**Max**−**Min**) across adaptation methods far exceeds that between StdPretrain and AdvPretrain, confirming adaptation as the stronger, more consistent factor.

3) Interaction between Pretraining and Adaptation

Setup. We jointly vary the pretraining mechanism and downstream adaptation, the latter adding PromptT and PrefixT on ViT-B only (to avoid Swin-B underfitting). `TriXfer` AEs are generated across multiple surrogate datasets (Fig. 14). We quantify variability via the ASR range (Max−Min).

Result. The impact of pretraining mechanisms on downstream robustness is relatively limited compared to adaptation methods. Across Swin-B and ViT-B, pretraining-induced ASR differences are small (0.03%–18%) versus much larger adaptation-induced variation (15%–63%). Thus, the choice of adaptation strategy has a greater impact on downstream robustness than the pretraining mechanism.

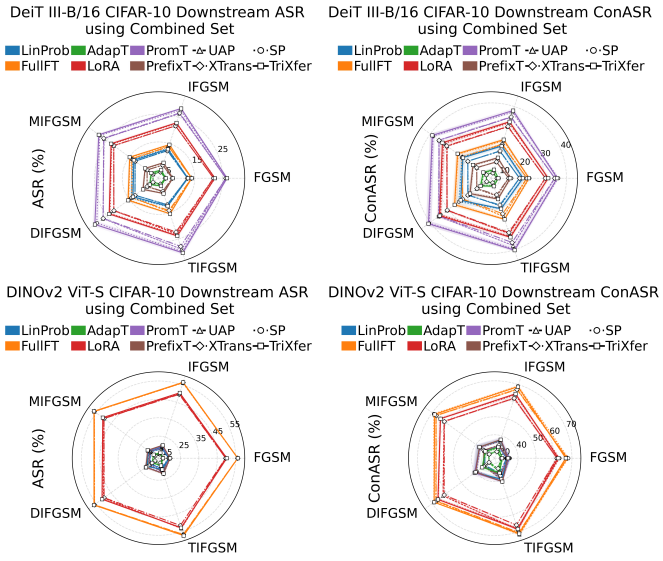


Fig. 17. Comparison with transfer attack baselines (Combined-CAW → CIFAR-10). Shading shows variability across four surrogate architectures.

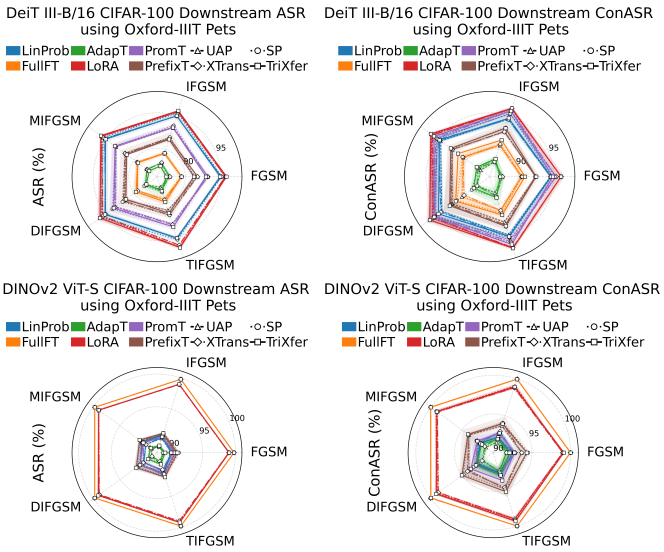


Fig. 18. Comparison with transfer attack baselines (Oxford-IIIT Pets → CIFAR-100). Shading shows variability across four surrogate architectures.

Takeaway of RQ2:

- *Downstream model robustness under transfer attacks is primarily determined by the downstream adaptation strategy rather than the ViT backbone pretraining mechanism.*
- *Although adversarial pretraining can induce modest changes in downstream robustness, its effects are limited and unstable across different backbone and adaptation settings.*
- *In contrast, downstream adaptation consistently induces much larger ASR variations and emerges as the dominant factor controlling downstream vulnerability.*
- *Robustness learned during pretraining does not reliably transfer to downstream tasks, highlighting robustness-aware transfer learning as a promising defense direction.*

F. Comparison with Transfer Attack Baselines

As the first work on untargeted triple-cross transfer attacks, we compare `TriXfer` against surrogate-based attack

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

TABLE III
SURROGATE-TO-VICTIM LABEL ALIGNMENT AXIS.

Level	Degree	Name	Mapping type
L1	Strong	Exact Match	1:1, lexically identical
L2	High	Alias Match	1:1, synonym
L3	Moderate	Many-to-one Concept	many:1, onto exact coarse concept
L4	Mild	Fine-grained Partial	concrete fine counterpart only
L5	Weak	Abstract Semantic Match	only a distant/abstract relative
L6	Zero	No Counterpart	no alias, no semantic relation

TABLE IV
CLASS-WISE TRIXFER GAIN BY ALIGNMENT LEVEL.

Alignment	Surrogate	Victim	ViT-B	Swin	DeiT	DINOv2	Avg
Foundation ASR							
L1 Exact	Oxford-37	ImageNet	+8.3	+19.6	+26.0	+13.4	+16.8
	CAW-306	ImageNet	+19.7	+31.2	+39.0	+24.9	+28.7
L4 Fine	Oxford-37	ImageNet	+9.7	+8.9	+15.3	+6.3	+10.1
	STL-10	ImageNet	+8.2	+13.8	+17.7	+10.6	+12.6
	CINIC-10	ImageNet	+12.4	+25.0	+40.9	+17.9	+24.0
L5 Abstract	STL-10	ImageNet	+3.6	+21.4	+24.8	+23.4	+18.3
	CINIC-10	ImageNet	+20.7	+27.8	+32.2	+19.8	+25.1
Downstream ASR							
L1 Exact	STL-10	CIFAR-10	+13.1	+18.2	+25.2	+34.3	+22.7
	CINIC-10	CIFAR-10	+22.5	+30.1	+43.8	+45.8	+35.6
L2 Alias	STL-10	CIFAR-10	+7.5	+20.0	+24.6	+44.4	+24.1
L3 Many to 1	Oxford-37	CIFAR-10	+17.7	+27.2	+22.1	+41.4	+27.1
	CAW-306	CIFAR-10	+9.5	+14.8	+13.6	+23.2	+15.3
L5 Abstract	Oxford-37	CIFAR-100	+2.7	+4.0	+5.0	+9.0	+5.2
Downstream ConASR							
L1 Exact	STL-10	CIFAR-10	+25.8	+22.5	+42.1	+39.9	+32.6
	CINIC-10	CIFAR-10	+30.4	+27.8	+47.4	+46.1	+37.9
L2 Alias	STL-10	CIFAR-10	+14.4	+19.0	+37.1	+52.4	+30.7
L3 Many to 1	Oxford-37	CIFAR-10	+21.4	+29.4	+31.6	+47.0	+32.4
	CAW-306	CIFAR-10	+11.2	+18.4	+17.0	+26.4	+18.2
L5 Abstract	Oxford-37	CIFAR-100	+3.9	+5.0	+6.0	+9.4	+6.1

baselines to validate our method. Unlike evaluations that assume target-domain samples, our AEs are crafted from clean surrogate samples and evaluated on black-box foundation and downstream models. We implement three baselines, SP [9], cross-domain UAP [42] (universal perturbation across samples and domains), and XTransfer [43] (cross-dataset universal perturbation), each combined with five perturbation operators.

1) Transfer Attacks under Hierarchical Overlap

Setup. Under *hierarchical overlap*, where surrogate fine-grained subclasses align with downstream labels only at the concept level, we use CIFAR-10/100 downstream and OxfordPets/Combined-CAW surrogates (Figs. 15, 17, 18). The shadowed region in the plot shows the variability across surrogates (ConvNeXtV2, IncepNeXt, MobViT, and EffViT).

Result. `TriXfer` surpasses all three baselines across perturbation operators, adaptations, and backbones. Gains are largest on CIFAR-10 with OxfordPets, e.g., $+2.31\%/+3.95\%/+4.13\%$ ASR and $+2.92\%/+3.68\%/+3.81\%$ ConASR over SP/UAP/XTransfer under the ConvNeXtV2 surrogate; remain positive on Combined-CAW, with the largest improvement under IncepNeXt: averaged $+0.16\%/+1.04\%/+1.15\%$ in ASR and $+0.74\%/+3.11\%/+3.38\%$ in ConASR over SP/UAP/XTransfer. On CIFAR-100, ASR for all methods already exceeds 90%, leaving little headroom. `TriXfer` still surpasses every baseline, with the largest gains under the EffViT surrogate: $+0.09\%/+0.17\%/+0.19\%$ in ASR and $+0.45\%/+0.38\%/+0.33\%$ in ConASR over SP/UAP/XTransfer. The advantage is consistently positive and largest over UAP and XTransfer, showing `TriXfer` adds value even where baseline transfer is already strong.

2) Transfer Attacks under Non-Hierarchical Overlap

Setup. Under *non-hierarchical overlap*, where surrogate and downstream labels share the same granularity, we use the STL-10 surrogate and CIFAR-10 downstream (Fig. 16).

Result. `TriXfer` consistently improves over baselines, most on the EffViT surrogate and for iterative operators. Averaged over operators, adaptations, and backbones (DeiT, DINO), it raises ASR from $35.50\%/35.22\%/35.05\%$ (SP/UAP/XTransfer) to 35.79% and ConASR from $44.92\%/44.54\%/44.30\%$ to 45.15%. These gains are smaller than under hierarchical overlap, since aligned labels make transfer easier, yet improvements over UAP and XTransfer stay stable. This shows `TriXfer` helps most in challenging cross-label-space tasks, not trivial label alignment.

G. Class-wise Analysis under Different Alignment Levels

To analyze attack effectiveness at a granular level, particularly under challenging cases such as many-to-one concept compression or abstract semantic mapping, we decompose performance along a six-level alignment axis (Table III), reporting class-wise ASR and ConASR by level (Table IV). Each per-victim value averages over surrogate classes at that level and six adaptation methods.

1) Surrogate-to-Pretrain Domain Alignment.

Result. Against the foundation domain, surrogate labels align at L1 (exact), L4 (fine-grained partial), and L5 (abstract). `TriXfer` yields substantial gains over the clean baseline at every level: L1 ($+16.8\%$ with OxfordPets, $+28.7\%$ with CAW), L4 ($+10.1\%/+12.6\%/+24.0\%$ with OxfordPets/STL/CINIC), and L5 ($+18.3\%/+25.1\%$ with STL/CINIC). The gain does not drop sharply as alignment weakens.

2) Surrogate-to-Downstream Domain Alignment.

Result. From L1 to L3 against CIFAR-10, effectiveness is maintained, L1 (STL, $+22.7\%$ ASR/$+32.6\%$ ConASR), L2 (STL, $+24.1\%/+30.7\%$), L3 (OxfordPets, $+27.1\%/+32.4\%$), showing that even a 37:2 many-to-one compression (OxfordPets $\rightarrow$ {cat, dog}) does not dilute the attack. The lower L3 CAW gain ($+15.3\%/+18.2\%$) reflects an extreme 306:3 compression across three heterogeneous fine-grained domains, not alignment-level degradation. At L5 (OxfordPets $\rightarrow$ CIFAR-100 wild relatives, with no direct domestic counterpart), `TriXfer` still gains $+5.2\%/+6.1\%$ atop a high ($> 85\%$) clean baseline. At L6, such surrogate classes are excluded during evaluation rather than forced onto a meaningless mapping, preserving the correctness and integrity of the remaining concept alignments. In our evaluation, L6 arises only once: the `monkey` class in STL-10 has no close concept in CIFAR-10, yet it aligns at L4 with ImageNet macaque/baboon synsets, achieving a significant gain of $+12.6\%$, demonstrating that the class is attackable whenever a semantic counterpart exists.

Overall, `TriXfer` yields consistent gains, and effectiveness does not degrade monotonously with alignment difficulty.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

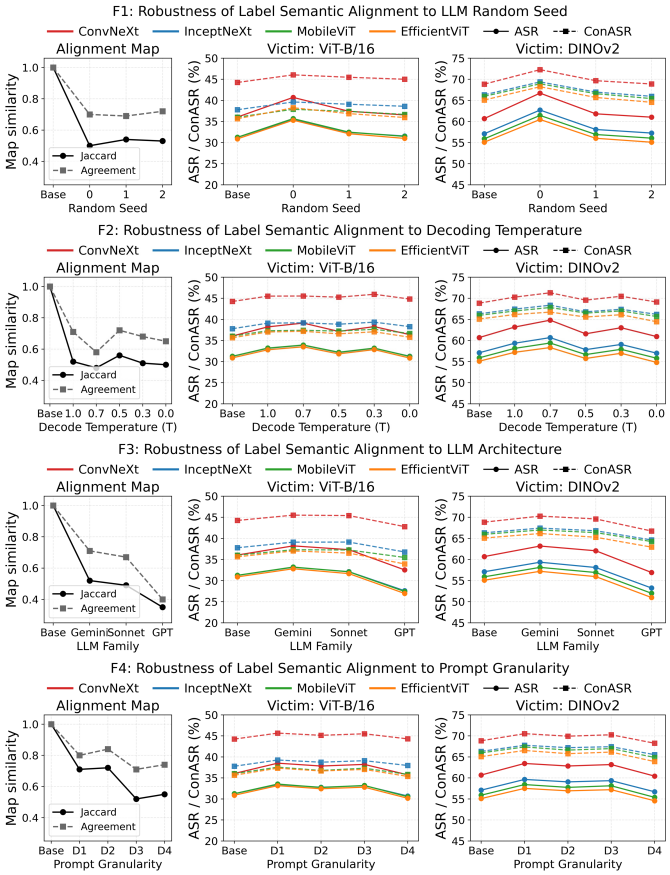


Fig. 19. Robustness to stochasticity in LLM-based concept alignment.

H. Robustness to LLM Stochasticity in Concept Alignment

The LLM-based label alignment involves stochastic components that may affect the generated concept mappings. We analyze robustness along four sensitivity factors (F1–F4) under the OxfordPets → CIFAR-10 setting (a hierarchical mismatch demanding more LLM-based semantic reasoning). For each factor, we jointly report attack performance, via ASR and ConASR, as well as mapping-level discrepancy, via Jaccard similarity (ratio of shared to total concept mappings between two alignment outputs) and label agreement rate (fraction of surrogate labels assigned the same concept across runs). We evaluate two victim backbones (ViT-B and DINOv2), with results averaged over four surrogate architectures and six adaptation methods (see Fig. 19).

1) F1: Robustness to LLM Random Seed.

We run the alignment pipeline with Gemini 2.5 Pro (T=1.0) across three seeds. Despite map-level variation (Jaccard 0.50–0.54, agreement 0.69–0.72), downstream attack metrics remain highly stable (ASR std $\leq 2.52\%$ and ConASR std $\leq 1.54\%$) across all configurations. This indicates that `TriXfer`'s attack performance is not sensitive to run-to-run stochasticity.

2) F2: Robustness to LLM Decoding Temperature.

We sweep $T \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$ with Gemini 2.5 Pro, running three seeds at each temperature. Although higher temperatures raise mapping-level diversity (Jaccard as low as 0.48), attack metrics remain stable throughout: ASR std $\leq 1.36\%$ and ConASR std $\leq 0.77\%$.

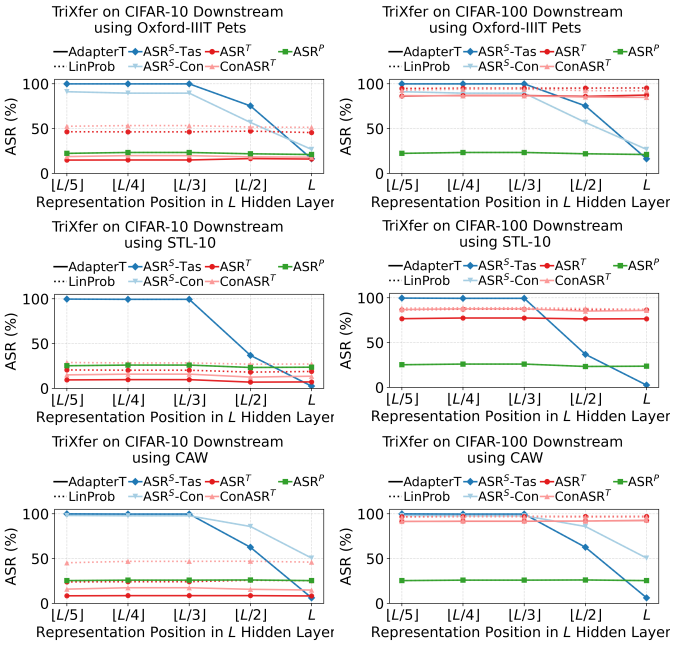


Fig. 20. Ablation on the hidden-representation layer index l used in $\mathcal{L}_{hid}$.

3) F3: Robustness to LLM Architecture.

We compare Gemini 2.5 Pro, Claude Sonnet 4.6, and GPT-4.1 ($T = 1.0$, three seeds each). Cross-LLM map similarity varies more than within a single LLM (Jaccard 0.35–0.52), with GPT producing the most divergent mapping. Nonetheless, the worst-case downstream impact remains modest: at most 4.10% in ASR and 2.15% in ConASR relative to the baseline. Gemini and Sonnet stay closer to the baseline, with ASR/-ConASR differences within $2.49\%/1.43\%$ and $1.37\%/1.34\%$, respectively. The baseline alignment employed in our work is derived from the cross-validation of various LLMs, making the alignment results more conservative and reliable.

4) F4: Robustness to LLM Prompt Granularity.

We vary the prompt along two independent controls: the *concept-group structure* (Specified or Free) and *hallucination filtering* (Filter vs. no Filter). Their combination gives four modes: D1 (Specified + Filter), D2 (Specified, no Filter), D3 (Free + Filter), and D4 (Free, no Filter). Even the least-constrained mode, D4, keeps ASR^T within 0.72% of the baseline, with std $\leq 1.20\%$ across all modes.

I. Ablation Studies

We ablate `TriXfer`'s key hyperparameters, reporting task-level and concept-level ASR on the surrogate model (ASR^S-Tas, ASR^S-Con), ASR on the pretrained ViTs (ASR^P), and both conventional and conditional ASR on the downstream model (ASR^T, ConASR^T), across the transfer attack pipeline.

1) Ablation on Hidden Representation Layer

Setup. We first vary the hidden layer l used by the hidden representation loss $\mathcal{L}_{hid}$ in Eq. (17) over $\{\lfloor L/5 \rfloor, \lfloor L/4 \rfloor, \lfloor L/3 \rfloor, \lfloor L/2 \rfloor, L\}$ (early to final), with the surrogate fixed to IncepNeXt, downstream models adapted by AdapterT and LinearProb, and DIFGSM as the operator.

Result. Across CIFAR-10 and CIFAR-100, middle-early layers consistently outperform very early and final hidden

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

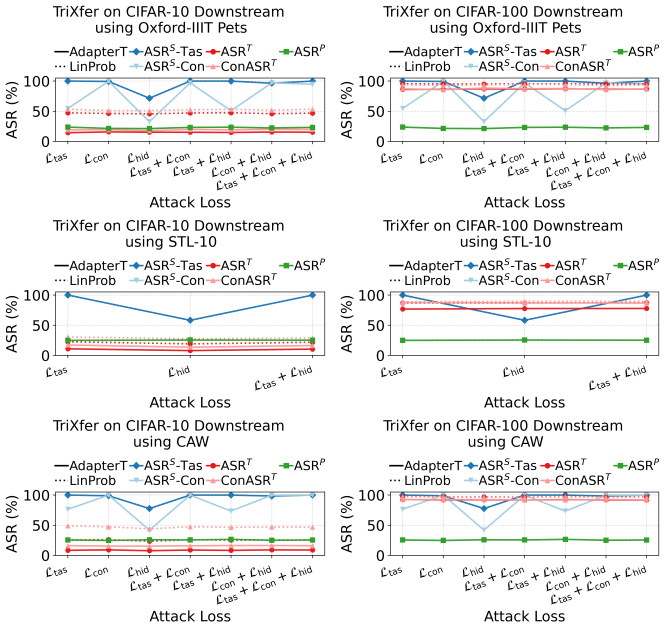


Fig. 21. Ablation on the multi-level attack-loss terms ($\mathcal{L}_{tas}$, $\mathcal{L}_{con}$, $\mathcal{L}_{hid}$) and their combinations.

layers (see Fig. 20). In particular, selecting a layer at approximately one-third depth yields the strongest and most stable performance across surrogate, pretrained, and downstream models, with downstream ASR^T and ConASR^T consistently maximized. By contrast, late-layer representations tend to overfit the surrogate task, while early layers are overly generic, resulting in weaker downstream transfer. Overall, middle-early representations strike the best balance between semantic relevance and cross-model invariance. We adopt $l = \lfloor L/3 \rfloor$ as the default hidden layer for $\mathcal{L}_{hid}$ in all experiments.

2) Ablation on Attack Loss Terms

Setup. We then ablate the attack loss by comparing individual terms and combinations of task-level loss $\mathcal{L}_{tas}$, concept-level loss $\mathcal{L}_{con}$, and hidden loss $\mathcal{L}_{hid}$ (see Fig. 21).

Result. Using a single loss term is insufficient to achieve strong and consistent transferability across evaluation stages. While $\mathcal{L}_{tas}$ alone yields high surrogate task-level ASR, it results in relatively lower downstream ASR^T, indicating limited transferability. Similarly, $\mathcal{L}_{hid}$ alone improves representation-level transfer but is unstable across datasets. Under hierarchical overlap, $\mathcal{L}_{con}$ contributes significantly more to increasing transferability across label spaces compared to $\mathcal{L}_{tas}$. The joint optimization of all three losses consistently maximizes downstream ASR^T and ConASR^T across both downstream datasets, while maintaining the best surrogate ASR and strong ASR on pretrained ViTs. Thus, these results show that effective transfer attacks require jointly optimizing multiple objectives.

3) Ablation on Loss Weights

Setup. We sweep the loss weights $(\lambda_{tas}, \lambda_{con}, \lambda_{hid})$ over the probability simplex ($\lambda_{tas}+\lambda_{con}+\lambda_{hid} = 1$, step size 0.25), yielding 15 configurations under hierarchical overlap and 5 under non-hierarchical overlap (where $\lambda_{con} = 0$). Experiments cover hierarchical surrogate datasets (Oxford, CAW) and non-hierarchical surrogate (STL-10), with two adaptation methods and two downstream datasets (see Fig. 22. and Fig. 23).

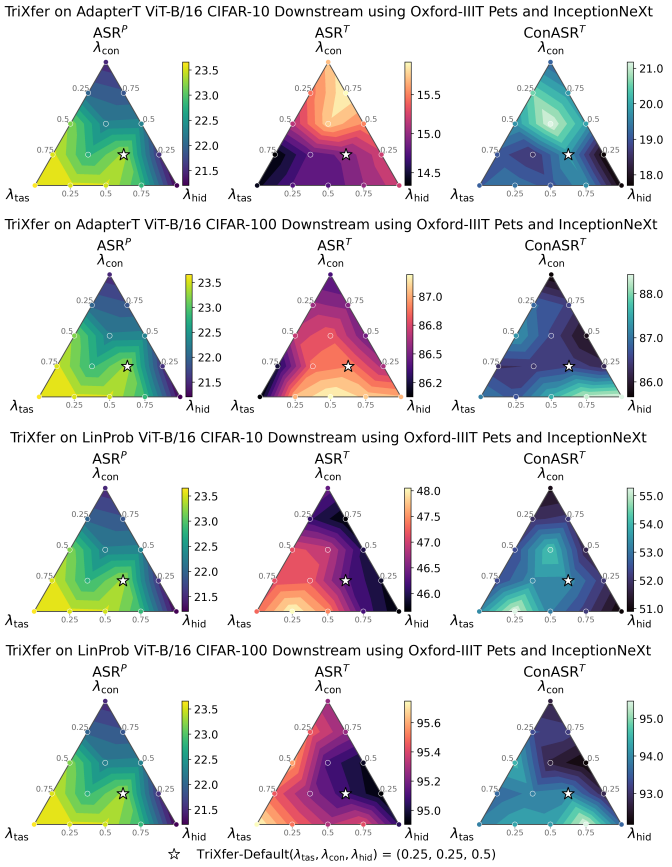


Fig. 22. Ablation on the loss weights $(\lambda_{tas}, \lambda_{con}, \lambda_{hid})$ under hierarchical.

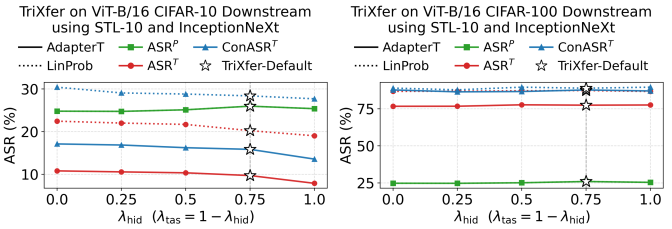


Fig. 23. Ablation on the loss weights $(\lambda_{tas}, \lambda_{hid})$ under non-hierarchical.

Result. The default weights follow a principled two-level selection criterion. (i) We prioritize configurations with high ASR on the pretrained foundation model (ASR^P), since ConASR on the downstream model (ConASR^T) is evaluated only on AEs that successfully fool the foundation, a larger conditional pool yields more reliable downstream estimates. (ii) Among configurations with comparable ASR^P, we select the one with the best overall downstream performance across various settings (surrogate datasets, downstream datasets, and adaptation methods). Under hierarchical overlap, the default $(0.25, 0.25, 0.5)$ ranks among the higher-ASR^P configurations while achieving competitive downstream ASR and ConASR across most settings. Under non-hierarchical overlap, λ_{con} is deactivated and the default $(0.25, 0, 0.75)$ similarly maximizes ASR^P. Since our evaluation spans a wide range of victim backbones, adaptation methods, and surrogate datasets, we adopt consistent default weights across all experiments to ensure fair comparison, rather than tuning per configuration.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

TABLE V
COMPARISON OF CONCEPT ALIGNMENT AND HARDLABEL RELABELING.

Surrogate	Adaptation	ASR			ConASR		
		Clean	RnR	TriXfer	Clean	RnR	TriXfer
OxfordPets-37	LinProb	22.6	41.7	41.8	24.6	49.9	50.7
	FullFT	25.8	57.9	58.0	32.4	66.2	66.2
	Adapter	4.5	16.4	16.9	4.9	23.6	24.6
	LoRA	2.8	33.7	34.1	5.4	44.3	45.0
	PromptT	36.9	57.6	58.0	39.5	65.3	66.1
	PrefixT	11.6	33.6	34.1	16.7	43.8	44.9
STL-10	LinProb	7.6	23.9	24.1	10.2	35.7	36.4
	FullFT	19.4	48.1	48.2	26.0	55.8	55.1
	Adapter	2.4	16.3	16.4	4.0	27.8	28.9
	LoRA	5.8	31.3	31.5	10.1	41.2	41.7
	PromptT	24.3	46.2	46.5	29.1	53.1	53.4
	PrefixT	6.8	28.5	28.7	17.4	41.7	42.2

J. Comparison of Concept-Level and Hard-Label Alignment

TriXfer aligns the surrogate in the shared concept space $\mathcal{S}_\cap$ (§IV-A), of which relabeling the surrogate with the victim's hard labels (RnR) is the special case when the surrogate-to-victim relation is exactly L3 many-to-one (see Table III).

1) Empirical Comparison with Hard-Label Relabeling

Setup. We analyze RnR in two settings: OxfordPets-37 (hierarchical) and STL-10 to CIFAR-10 (non-hierarchical). As RnR has no surrogate-to-foundation mapping, we filter its AEs with TriXfer's concept alignment to make ConASR comparable. Each value in Table V is averaged over six backbones, two surrogates, and two operators.

Result. Despite RnR using the exact victim labels, TriXfer exceeds RnR in nearly all settings. The gains are largest on OxfordPets under parameter-efficient TL, whereas on STL-10, the more favorable case for RnR since its label space already matches the victim's, TriXfer maintained a consistent and small advantage. Thus, the benefit of concept-level alignment grows with the degree of label-space compression.

2) Generality of Concept-Level Alignment

Crucially, concept-level alignment is more general. It (i) spans the full alignment spectrum, including one-to-many surrogate-to-foundation relations where no single foundation hard label can relabel the coarser surrogate; (ii) aligns to the overlap of both victim label spaces, so the same AEs jointly transfer to F^P and F^T, enabling our study of vulnerability propagation, which a single-victim-space surrogate cannot; and (iii) extends to any downstream label space by swapping the concept alignment at attack time, without retraining a victim-specific surrogate. This establishes that TriXfer is necessary beyond simple hard-label alignment.

VI. RELATED WORK

Cross-Model Transfer. Most transfer attack studies focus on cross-model architectural transferability, including early optimization and single-step methods [44], [53], as well as later approaches that enhance transferability by mitigating surrogate overfitting via gradient regularization [9], [10], [17], [46], [48], [54]–[56], representation-level objectives [8], [57], [58], or input diversification [18], [47], [59]–[61]. Except above single-surrogate methods, ensemble-surrogate methods have also been explored [16], [62], [63]. While ensembles often improve transferability, their high computational cost makes single-surrogate attacks appealing due to efficiency and competitive performance [64]. Empirically, cross-model transferbility has been demonstrated across early CNNs (e.g., LeNet, AlexNet, VGG) [44], [53], [55], [65]–[68], deep CNNs (e.g., GoogLeNet, Inception, ResNet, DenseNet, MobileNet, and EfficientNet) [11], [46], [48], [54], [57], [67]–[70], ViTs [11], [14], [15], [17], [58], and vision cloud [9], but still under the same dataset and label assumptions. Cross-model transfer does not account for scenarios in which surrogate and victim models differ in data distribution and label semantics.

Cross-Dataset Transfer. Another line studies cross-dataset transfer under fixed or closely related model architectures, e.g., using a pretrained model as the surrogate to attack its fine-tuned counterpart [19], [20], [71], [72], where surrogate and victim share the same backbone and even pretrained representations. Prior methods either craft universal perturbations on surrogate models and apply them to victim-domain data [42], [43], [67], [73], [74], which avoids victim-data access but lacks cross-domain semantic alignment, or generate AEs from clean victim-domain data using victim-domian labels [68] or hidden features [20], [75], [76]. However, existing cross-domain transfer attacks mainly demonstrate transferability across CNNs (e.g., AlexNet, VGG, ResNet, GoogLeNet, DenseNet) [67], [73], [74], [76], and can be divided into (i) intra-dataset transfer, where surrogate and victim models are trained on the same dataset family with different splits or variants [53], [62], [65], [68], and (ii) inter-dataset transfer, where models are trained on distinct datasets, yet using data-rich surrogates to attack smaller victims [20], [67], [68], [73], [74], [76]. These cross-domain transfer attacks rely on strong assumption, such as access to white-box knowledge of the victim backbone or victim-domain clean data between surrogate and victim tasks.

Cross-Label Space Transfer. Cross-label space transfer is far less studied. The only work [77] crafts *targeted* AEs from victim-domain clean data on a surrogate model using aligned target labels in the surrogate domain. It builds a correspondence for hard label alignment and optimizes targeted attack objectives [8], misleading black-box models with mismatched label spaces. However, its reliance on victim-domain clean data and predefined targets limits extension to untargeted attacks, and transfer was only validated among CNNs and from large surrogate to smaller victim domains.

Our Novelty. Most prior evidence relies on simplifying assumptions, identical datasets or label spaces, or access to clean victim-domain inputs, that often fail in the downstream adaptation of modern foundation models. We instead study an unexplored triple-cross adversary, where the pretrained ViT and downstream classifier are both black boxes, and AEs are crafted solely from clean surrogate data despite concurrent model, dataset, and label-space mismatches.

VII. CONCLUSION

We proposed `TriXfer`, the first triple-cross transfer attack that jointly addresses cross-model, cross-dataset, and cross-label space mismatches via label concept alignment and a multi-level attack objective. `TriXfer` reveals persistent

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

black-box vulnerability inheritance in ViT downstream adaptation regardless of pretrained backbones under a dual black-box setting, motivating robustness-aware adaptation strategies.

REFERENCES

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[2] A. Vaswani, "Attention is all you need," in *Annual Conference on Neural Information Processing Systems*, 2017.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021, pp. 8748–8763.

[5] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, 2021, pp. 8821–8831.

[6] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.

[7] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 133–16 142.

[8] N. Inkawhich, K. Liang, L. Carin, and Y. Chen, "Transferable perturbations of deep feature distributions," in *International Conference on Learning Representations*, 2020.

[9] Y. Mao, C. Fu, S. Wang, S. Ji, X. Zhang, Z. Liu, J. Zhou, A. X. Liu, R. Beyah, and T. Wang, "Transfer attacks revisited: A large-scale empirical study in real computer vision settings," in *IEEE Symposium on Security and Privacy*, 2022, pp. 1423–1439.

[10] Y. Zhang, S. Hu, L. Y. Zhang, J. Shi, M. Li, X. Liu, and H. Jin, "Why does little robustness help? a further step towards understanding adversarial transferability," in *Symposium on Security and Privacy*, 2024.

[11] K. Mahmood, R. Mahmood, and M. Van Dijk, "On the robustness of vision transformers to adversarial examples," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7838–7847.

[12] A. Joshi, G. Jagatap, and C. Hegde, "Adversarial token attacks on vision transformers," *arXiv preprint arXiv:2110.04337*, 2021.

[13] Y. Fu, S. Zhang, S. Wu, C. Wan, and Y. Lin, "Patch-fool: Are vision transformers always robust against adversarial perturbations?" in *International Conference on Learning Representations*, 2022.

[14] Z. Qin, Y. Fan, Y. Liu, L. Shen, Y. Zhang, J. Wang, and B. Wu, "Boosting the transferability of adversarial attacks with reverse adversarial perturbation," in *Annual Conference on Neural Information Processing Systems*, 2022, pp. 29 845–29 858.

[15] M. Naseer, K. Ranasinghe, S. Khan, F. Khan, and F. Porikli, "On improving adversarial transferability of vision transformers," in *International Conference on Learning Representations*, 2022.

[16] B. Chen, J. Yin, S. Chen, B. Chen, and X. Liu, "An adaptive model ensemble adversarial attack for boosting adversarial transferability," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4489–4498.

[17] J. Zhang, Y. Huang, W. Wu, and M. R. Lyu, "Transferable adversarial attacks on vision transformers with token gradient regularization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 415–16 424.

[18] D. Ming, P. Ren, Y. Wang, and X. Feng, "Boosting the transferability of adversarial attack on vision transformer with adaptive token tuning," in *Annual Conference on Neural Information Processing Systems*, 2024, pp. 20 887–20 918.

[19] Y. Ban and Y. Dong, "Pre-trained adversarial perturbations," *Advances in neural information processing systems*, vol. 35, pp. 1196–1209, 2022.

[20] W. Zheng, X. Ma, H. Huang, Z. Wu, and Y.-G. Jiang, "Downstream transfer attack: Adversarial attacks on downstream models with pre-trained vision transformers," *arXiv preprint arXiv:2408.01705*, 2024.

[21] S. Xia, W. Yang, Y. Yu, X. Lin, H. Ding, L. Duan, and X. Jiang, "Transferable adversarial attacks on sam and its downstream models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 545–568, 2024.

[22] H. Touvron, M. Cord, and H. Jégou, "Deit iii: Revenge of the vit," in *European conference on computer vision*. Springer, 2022, pp. 516–533.

[23] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[24] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 32–42.

[25] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, and L. Yuan, "Tinyvit: Fast pretraining distillation for small vision transformers," in *European conference on computer vision*. Springer, 2022, pp. 68–85.

[26] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*, 2019, pp. 2790–2799.

[27] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[28] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European conference on computer vision*. Springer, 2022, pp. 709–727.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[30] Y. Mo, D. Wu, Y. Wang, Y. Guo, and Y. Wang, "When adversarial training meets vision transformers: Recipes from training to architecture," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 599–18 611, 2022.

[31] W. Yu, P. Zhou, S. Yan, and X. Wang, "Inceptionnext: When inception meets convnext," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5672–5683.

[32] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *International Conference on Learning Representations*, 2022.

[33] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 420–14 430.

[34] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3498–3505.

[35] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *IEEE/CVF International Conference on Computer Vision Workshops*, 2013, pp. 554–561.

[36] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *preprint arXiv:1306.5151*, 2013.

[37] fast.ai, "Imagewoof: A subset of imagenet for image classification," https://github.com/fastai/imagenette, 2019.

[38] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[39] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "Cinic-10 is not imagenet or cifar-10," *arXiv preprint arXiv:1810.03505*, 2018.

[40] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 2009.

[41] J. Howard, "Imagenette: A subset of 10 classes from imagenet," 2019. [Online]. Available: https://github.com/fastai/imagenette

[42] M. M. Naseer, S. H. Khan, M. H. Khan, F. Shahbaz Khan, and F. Porikli, "Cross-domain transferability of adversarial perturbations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[43] H. Huang, S. M. Erfani, Y. Li, X. Ma, and J. Bailey, "X-transfer attacks: Towards super transferable adversarial attacks on clip," in *International Conference on Machine Learning*, 2025.

[44] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[45] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[46] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[47] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity,"

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3721888

in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[48] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12–21.

[49] N. Tripuraneni, M. Jordan, and C. Jin, "On the theory of transfer learning: The importance of task diversity," *Advances in neural information processing systems*, vol. 33, pp. 7852–7862, 2020.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[51] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.

[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision*, vol. 115, no. 3, 2015.

[53] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[54] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," in *International Conference on Learning Representations*, 2020.

[55] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with resnets," *arXiv preprint arXiv:2002.05990*, 2020.

[56] M. Gubri, M. Cordy, M. Papadakis, Y. L. Traon, and K. Sen, "Lgv: Boosting adversarial example transferability from large geometric vicinity," in *European Conference on Computer Vision*, 2022, pp. 603–618.

[57] Q. Huang, I. Katsman, H. He, Z. Gu, S. Belongie, and S.-N. Lim, "Enhancing adversarial example transferability with an intermediate level attack," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4733–4742.

[58] Z. Wei, J. Chen, M. Goldblum, Z. Wu, T. Goldstein, and Y.-G. Jiang, "Towards transferable adversarial attacks on vision transformers," in *AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 68–76.

[59] Q. Li, Y. Guo, and H. Chen, "Practical no-box adversarial attacks against dnns," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 849–12 860, 2020.

[60] X. Wang, X. He, J. Wang, and K. He, "Admix: Enhancing the transferability of adversarial attacks," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 158–16 167.

[61] Z. Yuan, J. Zhang, and S. Shan, "Adaptive image transformations for transfer-based adversarial attack," in *European Conference on Computer Vision*. Springer, 2022, pp. 1–17.

[62] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *International Conference on Learning Representations*, 2017.

[63] Y. Li, S. Bai, Y. Zhou, C. Xie, Z. Zhang, and A. Yuille, "Learning transferable adversarial examples via ghost networks," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 11 458–11 465.

[64] N. A. Lord, R. Mueller, and L. Bertinetto, "Attacking deep networks with surrogate-based adversarial black-box methods is easy," in *International Conference on Learning Representations*, 2022.

[65] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[66] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint arXiv:1611.02770*, 2016.

[67] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1765–1773.

[68] Y. Zhang, Y. Song, K. Bai, and Q. Yang, "Cross-domain cross-architecture black-box attacks on fine-tuned models with transferred evolutionary strategies," in *ACM International Conference on Information & Knowledge Management*, 2022, pp. 2661–2670.

[69] M. Naseer, S. H. Khan, S. Rahman, and F. Porikli, "Task-generalizable adversarial attack based on perceptual metric," *arXiv preprint arXiv:1811.09020*, 2018.

[70] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks," in *USENIX Security Symposium*, 2019, pp. 321–338.

[71] B. Wang, Y. Yao, B. Viswanath, H. Zheng, and B. Y. Zhao, "With great training comes great vulnerability: Practical attacks against transfer learning," in *USENIX Security Symposium*, 2018, pp. 1281–1297.

[72] Y. Na, J. H. Kim, K. Lee, J. Park, J. Y. Hwang, and J. P. Choi, "Domain adaptive transfer attack-based segmentation networks for building extraction from aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 5171–5182, 2020.

[73] K. R. Mopuri, U. Garg, and R. V. Babu, "Fast feature fool: A data independent approach to universal adversarial perturbations," *arXiv preprint arXiv:1707.05572*, 2017.

[74] K. R. Mopuri, A. Ganeshan, and R. V. Babu, "Generalizable data-free objective for crafting universal adversarial perturbations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2452–2465, 2018.

[75] Q. Zhang, X. Li, Y. Chen, J. Song, L. Gao, Y. He *et al.*, "Beyond imagenet attack: Towards crafting adversarial examples for black-box domains," in *International Conference on Learning Representations*, 2022.

[76] Z. Li, W. Wu, Y. Su, Z. Zheng, and M. R. Lyu, "Cdta: a cross-domain transfer-based attack with contrastive learning," in *AAAI Conference on Artificial Intelligence*, 2023, pp. 1530–1538.

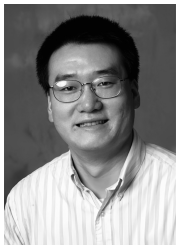
[77] N. Inkawhich, K. J. Liang, J. Zhang, H. Yang, H. Li, and Y. Chen, "Can targeted adversarial examples transfer when the source and target models have no label space overlap?" in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 41–50.



Mengdie Huang is a Postdoctoral Research Associate in the Department of Computer Science at Purdue University, West Lafayette, US. From 2022 to 2025, she was a Visiting Scholar in the same department at Purdue University. She received her Ph.D. degree in Cyberspace Security from Xidian University, Xi'an, China, in 2025, and M.S. degree in Electronic Information Engineering from Communication University of China, Beijing, China, in 2019. Her research interests are ML security, network security, and multimedia security.



Yingjun Lin is pursuing the M.S. degree in computer science at Georgia Institute of Technology. He received the B.S. degree in data science from Purdue University, West Lafayette, US, in 2024. His research interests include ML security, generative AI, LLM, intelligent agents, vision-language models, and computer vision.



Ninghui Li (Fellow, IEEE) is a Samuel D. Conte Professor of Computer Science at Purdue University. He received a Bachelor's degree from University of Science and Technology of China in 1993, and a Ph.D. in Computer Science from New York University in 2000. His research interests are in security and privacy. He is Editor-in-Chief for ACM TOPS since October 2020, and has served as Program Chair for several conferences, including ACM CCS, ESORICS, ACM ASIACCS, ACM SACMAT, etc. He served as Chair of ACM SIGSAC from 2017 to 2021. He is a fellow member of IEEE and ACM.



Elisa Bertino (Fellow, IEEE) is a Samuel D. Conte Distinguished Professor of Computer Science at Purdue University. She serves as director of the Purdue Cyberspace Security Lab, and leads research in data security and privacy. Before joining Purdue in 2004, she was a professor and department head with the Department of Computer Science and Communication, University of Milan. She has been a postdoc researcher with the IBM Research Laboratory in San Jose, and a visiting professor with Singapore Management University and Singapore National University. She is a fellow member of IEEE, ACM, and AAAS.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.