DetectViT: Test-time Backdoor Detection for Vision Transformers via Inter-Head Attention Discrepancy

Siquan Huang[1] **Yijiang Li**[2] **Xingfu Yan**[3,*] **Ningzhi Gao**[1] **Leyu Shi**[1] **Ying Gao**[1,*]

[1]South China University of Technology [2]University of California, San Diego

[3]South China Normal University

{cssiquanhuang, csgaonz, csshileyu}@mail.scut.edu.cn, gaoying@scut.edu.cn

yijiangli@ucsd.edu, xfyan78@163.com

Abstract

Vision Transformers (ViTs) now underpin the visual encoders of most multimodal models, but this success comes at a cost: relying on third-party pretrained checkpoints leaves these systems open to backdoor attacks. Existing test-time detection methods rely on either the predicted label or intermediate embeddings, limiting their applicability across diverse tasks and incurring substantial computational overhead. In this work, we study how backdoor triggers affect the multi-head attention mechanism of ViTs and identify two distinct anomalous patterns: *complete attention hijacking*, causing the attention to exhibit abnormally high inter-head spatial overlap, and *partial attention hijacking*, causing an abnormally large variance in the concentration of the attention distribution. Motivated by these findings, we propose **DetectViT**, a test-time backdoor detection method that requires neither training data, model outputs, nor any prior knowledge of the trigger. DetectViT quantifies the two hijacking phenomena via an inter-head consistency score and an inter-head entropy variance score, with thresholds estimated from a small out-of-distribution (OOD) calibration set drawn from an arbitrary source. Across four representative attacks, diverse architectures (DeiT, CLIP, and LLaVA), and downstream tasks (image classification and captioning), DetectViT substantially outperforms all baselines with as few as 64–128 OOD calibration samples, achieving a true positive rate of 100% and a false positive rate of 0.17% on BadCLIP. Notably, it introduces negligible additional overhead by relying solely on attention weights already computed during inference. Code is available at: `https://github.com/siquanhuang/DetectViT`.

1 Introduction

Vision Transformers (ViTs) [10, 47, 37] have emerged predominantly as the backbone for visual representation learning, serving as the de facto visual encoder in multimodal models [42, 38] and large vision-language models (LVLMs) [64, 31, 1, 55]. However, training ViTs from scratch has become increasingly prohibitive as model scales continue to grow [60, 7]. Practitioners therefore rely on third-party pretrained checkpoints, exposing their systems to backdoor attacks [5, 20, 22, 19, 14]. Specifically, an adversary can take advantage of this by injecting a backdoor directly into the pretrained model. This threat is further amplified by ViT-specific attacks that exploit the self-attention mechanism to achieve high attack success while remaining stealthy [35, 58, 61, 53].

Existing backdoor detection methods commonly require access to training data to estimate a reference distribution, enabling effective filtering of poisoned samples during the training phase [40, 43, 57].

*Corresponding authors: Ying Gao and Xingfu Yan.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Yet, this assumption is often unrealistic when practitioners obtain pretrained ViTs from third-party sources with no knowledge of the original training process. Moreover, backdoors embedded in pretrained checkpoints can survive downstream fine-tuning [23, 45, 30, 34, 58, 61], underscoring the need for test-time detection. To address this threat, existing test-time methods predominantly inspect the model's output on suspicious samples, either through predicted labels or intermediate embeddings. However, both paradigms suffer from fundamental limitations. Label-based methods [13, 9, 16, 32] are inherently restricted to classification tasks and cannot handle open-ended outputs such as image captioning or visual question answering, where no discrete label space exists. Embedding-based methods [18, 21] sidestep this restriction but are blind to attacks that target the classification head rather than the encoder [61], leaving such backdoors entirely undetected. Moreover, both paradigms typically rely on generating multiple augmented copies of each test sample [32, 16, 9, 21, 44], introducing substantial inference-time overhead. These limitations motivate a fundamentally different perspective: *Can we detect backdoor samples by directly probing the internal computation of the ViT itself, without any training data or model outputs?*

To address this question, we turn to the core component of ViTs, the multi-head attention mechanism, and ask: *What does a backdoor trigger do to attention?* While prior work [58, 61, 43] has shown that triggers can hijack heads' attention toward themselves, we go further by analyzing this hijacking through an *inter-head* lens. Specifically, on clean images, different attention heads naturally attend to spatially distinct regions through their separate representational subspaces, and exhibit broadly comparable concentration across heads, owing to their shared architecture and optimization objective. On backdoored images, however, we observe that triggers disrupt both this diversity and this balance, giving rise to two characteristics: **1) complete attention hijacking**, where a prominent, patch trigger drives nearly all heads to attend to the trigger region, yielding abnormally high inter-head spatial overlap; and **2) partial attention hijacking**, in which a stealthier blended trigger affects heads unevenly, leading to abnormally high variance in attention concentration across heads. Together, these two characteristics cover the two dominant trigger families (patch-based and blended-based), providing a principled signal for detecting backdoor without requiring training data or model outputs.

Motivated by these findings, we propose **DetectViT**, a test-time backdoor detection method by examining the attention matrix of ViTs. For each test sample, DetectViT extracts the CLS-to-patch attention distributions across all heads and layers, and computes two complementary scores: an inter-head consistency score $\mathcal{S}_{\text{cons}}$ that captures **complete attention hijacking**, and an inter-head entropy variance score $\mathcal{S}_{\text{var}}$ that captures **partial attention hijacking**. Detection thresholds are estimated using a small out-of-distribution (OOD) calibration set drawn from any readily available dataset. A test sample is flagged as backdoored if either score exceeds its corresponding threshold.

Extensive experiments on four representative attacks across diverse ViT-based architectures (DeiT, CLIP, LLaVA) and downstream tasks (image classification and image captioning) demonstrate that DetectViT substantially outperforms existing baselines. For example, on BadCLIP, DetectViT achieves a true positive rate of 100% at a false positive rate of only 0.17% using as few as 128 OOD calibration samples. Crucially, DetectViT introduces no additional inference latency since it uses attention weights already computed during inference, making it well-suited for real-world deployment. We summarize our main contributions below:

- We identify two anomalous attention phenomena induced by backdoors, i.e., **complete** and **partial attention hijacking**, characterized by abnormally high inter-head attention overlap and large inter-head variance in attention concentration. To our knowledge, this is the first systematic characterization of backdoor behavior through inter-head attention discrepancy.

- We propose **DetectViT**, a test-time detection method by capturing these two phenomena via an inter-head consistency score and an inter-head entropy variance score. **DetectViT** requires neither training data, model outputs, nor any prior knowledge of the trigger.

- Extensive experiments across diverse attacks, models, and tasks show that DetectViT achieves state-of-the-art performance with only a small OOD calibration set, remains robust to the source and size of the calibration data, and is highly effective against adaptive attacks, with negligible computation overhead.

2 Related Work

Vision Transformers. Vision Transformers [10] introduced the transformer architecture [49] to visual representation learning by treating an image as a sequence of fixed-size patches and processing them with multi-head self-attention. Subsequent variants have demonstrated strong performance across diverse vision tasks [33, 52, 27, 47, 29, 37]. Among them, DeiT [47] extended ViTs by introducing a knowledge distillation strategy that enables efficient training without massive datasets. Building on them, CLIP [42] proposed a vision-language model trained on large-scale image-text pairs via contrastive learning, inspiring a series of VLMs [26, 56]. LVLMs like LLaVA [31] further advanced this direction by connecting a pretrained ViT-based vision encoder to a large language model, enabling image captioning and visual question answering, with numerous follow-up works adopting similar architectures [64, 51, 28, 55, 46].

Backdoor Attacks on Vision Transformers. Backdoor attacks [14, 5] can manipulate the victim model to produce attacker-specified predictions on trigger-bearing inputs while behaving normally on clean data. Recent works [58, 61] have demonstrated that the self-attention mechanism of ViTs introduces vulnerabilities that can be exploited to craft more effective and stealthy backdoor attacks. BadViT [58] designs an optimized patch-based trigger that manipulates the attention to concentrate on the trigger region, achieving high attack success rates. TrojViT [61] optimizes a multiple-patch trigger on adaptive regions and takes a different approach by injecting the backdoor into the classification head rather than the encoder, making it harder to detect via encoder-level analysis. AGEB [15] proposes an attention gradient-based erosion strategy that selectively corrupts pixels in regions of maximal attention gradient. AIBA [53] follows TrojViT, optimizing a multiple-patch adversarial perturbation as the trigger.

Beyond classification, backdoor attacks have been extended to ViT-based VLMs. BadCLIP [30] targets the CLIP architecture by designing a trigger that manipulates both the visual and textual representations. More recently, BadVision [34] reveals a new and more far-reaching threat: by compromising only the pretrained vision encoder, an adversary can induce significant visual hallucinations in downstream LVLMs such as LLaVA [31] and MiniGPT-4 [64] without touching the language model. Since pretrained vision encoders are widely shared and reused across many LVLMs, a single poisoned encoder can propagate backdoor behaviors to numerous downstream models. This highlights the urgent need for effective detection approaches that operate at the vision encoder level.

Test-time Backdoor Samples Detection. Multiple methods detect backdoors requiring training data [12, 40, 43, 57], making them inapplicable to protect third-party pretrained models. Test-time detection methods inspect suspicious samples at inference time without requiring training data. The pioneering work STRIP [13] adds perturbations to test inputs and flags samples whose labels remain abnormally stable. TeCo [32] analyzes the consistency of hard labels under a series of image degradation transformations. Scale-Up [16] amplifies the test image by varying scaling factors and detects samples based on their label changes. PatchProc [9] processes images at the patch level and identifies backdoor samples via label shifts under patch-wise manipulations. However, these label-based methods are fundamentally restricted to classification tasks and cannot handle open-ended outputs like image captioning. Advanced embedding-based methods avoid this restriction by operating on vision encoder representations: DeDe [18] detects anomalous encoder mappings via a trained decoder; BackdoorIDS [21] clusters the embedding sequence under progressive masking to detect backdoor behavior, and SubspaceDetection [44] identifies poisoned CLIP images via shallow text-subspace alignment. However, these embedding-based methods are blind to backdoors injected into the classification head. Moreover, most existing test-time methods [32, 16, 9, 21, 44] rely on generating a large number of augmented variants per test sample, incurring substantial computational overhead at inference time.

In contrast, DetectViT addresses the above limitations by operating on the internal multi-head attention behavior of ViTs, without relying on model outputs, detecting different types of triggers, and generalizing across classification and multimodal tasks with negligible additional time overhead.

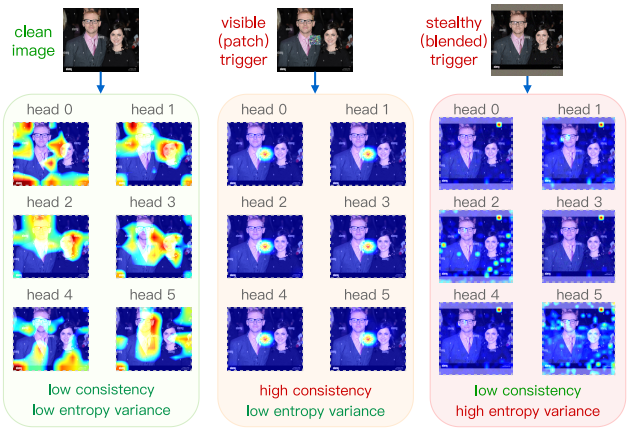
3 Motivation

The multi-head attention mechanism in ViTs is fundamentally designed to capture diverse and complementary visual representations: each head independently projects token embeddings into a

separate subspace, allowing the model to simultaneously analyze different aspects of a visual scene. As shown in Fig. 1a, for a clean image, this design naturally induces **attention diversity and balance**: as each head perceives the image through a distinct lens, different heads attend to spatially different regions [3]; meanwhile, since all heads share the same architectural capacity and are optimized under the same objective, their concentration of the attention distribution remains broadly comparable.

We argue that backdoor triggers disrupt this diversity and balance. While prior work [58, 61, 43] has shown that triggers hijack heads' attention, our analysis further reveals that this hijacking is far from consistent across heads; rather, it manifests in two distinct regimes depending on the perceptibility and spatial extent of the trigger, as empirically corroborated in Sec. 6 (Fig. 4):

1. Complete attention hijacking occurs when a prominent, localized trigger is present. The trigger pattern introduces an anomalously dominant local feature that overrides the learned diversity of the multi-head mechanism, forcing all attention heads to converge onto the trigger region regardless of their respective subspace projections. As visualized in Fig. 1b, the attention maps of all heads collapse onto a single small region corresponding to the trigger, exhibiting an inter-head spatial overlap that stands in sharp contrast to the diverse distributions observed on clean images.



(a) Clean image (b) Complete hijacking (c) Partial hijacking

Figure 1: Visualization of the head attention distribution over the CLS token of clean/backdoor images.

2. Partial attention hijacking characterizes the behavior of stealthy, low-visibility triggers, such as blended-based attacks, where a global pattern is superimposed at low intensity across the entire image. The backdoor is activated by the cumulative contribution of many faint, image-wide patches rather than any single dominant region; consequently, different heads respond to the trigger in fundamentally different ways, as shown in Fig. 1c: some heads are fully hijacked and attend broadly across nearly all trigger-bearing patches (e.g., heads 2 and 5), while others are only partially captured and concentrate narrowly on the most salient trigger patches (e.g., heads 0 and 3). This heterogeneous response produces a pronounced disparity in the concentration of the attention distribution across each head, whereas clean images yield the comparable concentration.

Taken together, these two forms of attention hijacking provide a principled and model-agnostic foundation for detecting backdoor samples across both visible and stealthy trigger modalities, requiring neither access to training data nor any assumption about the specific trigger pattern employed.

4 Method

We present **DetectViT**, a novel test-time framework for detecting backdoor samples in ViTs. The method exploits two complementary deviations in multi-head attention behavior induced by backdoor triggers: abnormal inter-head spatial overlap under complete hijacking, and abnormal inter-head concentration variance under partial hijacking. The overall pipeline is illustrated in Fig. 2.

4.1 Attention Distribution of Each Head over the CLS Token

Given an input image x, we pass it through the ViT encoder and extract the attention weights produced at each transformer layer. Formally, let L denote the total number of layers and H the number of attention heads per layer. At layer $\ell \in \{1, \ldots, L\}$, let $\mathbf{A}^{(\ell,h)} \in \mathbb{R}^{(N+1)\times(N+1)}$ denote the attention weight matrix for head h, where N is the number of patch tokens, and the additional token corresponds to the CLS token.

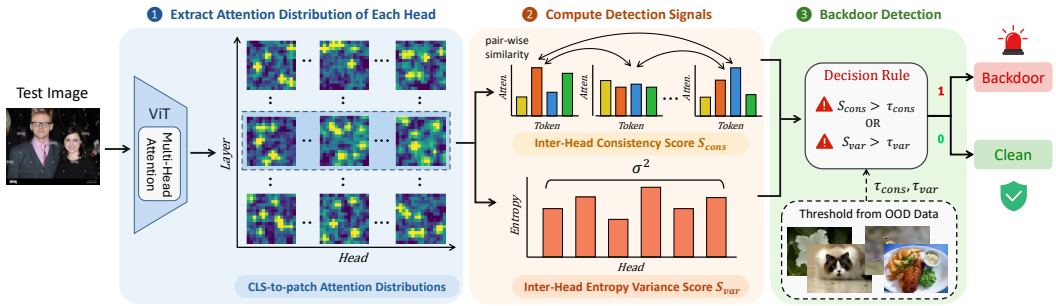


Figure 2: Overview of our proposed DetectViT.

We focus on the attention that the CLS token pays to each patch token, as it encodes the global semantic representation of the image. We extract the CLS-to-patch attention vector for each head:

$$\mathbf{a}^{(\ell,h)} = \mathbf{A}^{(\ell,h)}_{0,1:N} \in \mathbb{R}^N, \tag{1}$$

where the subscript 0 indexes the CLS token row and $1{:}N$ indexes the N patch token columns. We then normalize each vector to form a proper probability distribution:

$$\tilde{\mathbf{a}}^{(\ell,h)} = \frac{\mathbf{a}^{(\ell,h)}}{\sum_{i=1}^{N} a_i^{(\ell,h)}} \in \mathbb{R}^N. \tag{2}$$

The resulting $\tilde{\mathbf{a}}^{(\ell,h)}$ represents the normalized attention distribution of head h at layer ℓ over all patch tokens, and serves as the basis for the two detection signals described below.

4.2 Detection Signals

4.2.1 Inter-Head Consistency Score

To detect complete attention hijacking, wherein all heads converge onto the same trigger region, we quantify the resulting inter-head spatial overlap via the pairwise cosine similarity between the attention distributions of different heads within the same layer. For any pair of heads (h, h') at layer ℓ, the cosine similarity is:

$$\mathrm{sim}^{(\ell)}(h,h') = \frac{\tilde{\mathbf{a}}^{(\ell,h)} \cdot \tilde{\mathbf{a}}^{(\ell,h')}}{\|\tilde{\mathbf{a}}^{(\ell,h)}\|_2 \cdot \|\tilde{\mathbf{a}}^{(\ell,h')}\|_2}. \tag{3}$$

We then aggregate these into a layer-wise inter-head consistency score as the mean over all pairs:

$$c^{(\ell)} = \frac{1}{\binom{H}{2}} \sum_{h<h'} \mathrm{sim}^{(\ell)}(h,h'). \tag{4}$$

A high value of $c^{(\ell)}$ indicates that all heads are attending to the same spatial region, which is the hallmark of complete attention hijacking. The final consistency score for the sample is taken as the maximum across all layers:

$$\mathcal{S}_{\text{cons}} = \max_{\ell} \; c^{(\ell)}. \tag{5}$$

4.2.2 Inter-Head Entropy Variance Score

To detect partial attention hijacking, wherein different heads exhibit divergent degrees of attention concentration, we quantify the concentration of each head's attention distribution by Shannon entropy:

$$\mathcal{H}^{(\ell,h)} = -\sum_{i=1}^{N} \tilde{a}_i^{(\ell,h)} \log\left(\tilde{a}_i^{(\ell,h)}\right). \tag{6}$$

A low entropy value indicates that the head concentrates its attention on a small number of patches (i.e., highly focused), while a high entropy value indicates diffuse attention over many patches. Under

partial hijacking, different heads respond to the trigger with heterogeneous concentration, producing an abnormally high variance in entropy across heads within a layer, which we refer to as the inter-head entropy variance score:

$$v^{(\ell)} = \mathrm{Var}_h\Big(\mathcal{H}^{(\ell,h)}\Big) = \frac{1}{H}\sum_{h=1}^{H}\Big(\mathcal{H}^{(\ell,h)} - \bar{\mathcal{H}}^{(\ell)}\Big)^2, \tag{7}$$

where $\bar{\mathcal{H}}^{(\ell)} = \frac{1}{H}\sum_h \mathcal{H}^{(\ell,h)}$ is the mean entropy at layer ℓ. The final entropy variance score is:

$$\mathcal{S}_{\text{var}} = \max_{\ell} v^{(\ell)}. \tag{8}$$

4.3 Threshold Estimation via OOD Calibration

Instead of relying on in-distribution training data, we estimate detection thresholds from a small set of OOD samples that are presumed to be clean. In practice, we find that as few as $M = 64$ or 128 such samples suffice. For each calibration sample x_m, we compute both scores $\mathcal{S}_{\text{cons}}(x_m)$ and $\mathcal{S}_{\text{var}}(x_m)$. We then estimate the empirical mean and standard deviation of each score over the calibration set:

$$\mu_{\text{cons}} = \frac{1}{M}\sum_{m=1}^{M}\mathcal{S}_{\text{cons}}(x_m), \quad \sigma_{\text{cons}} = \mathrm{Std}_m(\mathcal{S}_{\text{cons}}(x_m)), \tag{9}$$

$$\mu_{\text{var}} = \frac{1}{M}\sum_{m=1}^{M}\mathcal{S}_{\text{var}}(x_m), \quad \sigma_{\text{var}} = \mathrm{Std}_m(\mathcal{S}_{\text{var}}(x_m)). \tag{10}$$

Since the scores of clean samples approximately follow a Gaussian distribution (verified in Appendix C.1), we apply the three-sigma rule to set the detection thresholds:

$$\tau_{\text{cons}} = \mu_{\text{cons}} + 3\sigma_{\text{cons}}, \qquad \tau_{\text{var}} = \mu_{\text{var}} + 3\sigma_{\text{var}}. \tag{11}$$

This ensures that the probability of a clean sample being falsely flagged as backdoor is less than 0.15% under the one-sided three-sigma rule, providing a conservative false positive rate.

Importantly, our method does not restrict the source of the OOD calibration samples. We empirically verify that thresholds estimated from a diverse range of OOD datasets yield consistently strong detection performance, including iNaturalist [17], Food101 [2], Oxford-IIIT Pet [41], Places365 [62], STL-10 [6], Flowers102 [39], and PASCAL VOC [11]. This flexibility makes the method practical in real-world scenarios where access to in-distribution clean data may be restricted.

4.4 Test-time Backdoor Sample Detection

Given a test sample x, we compute both detection scores and compare them against the estimated thresholds. The sample is classified as backdoored if either score exceeds its corresponding threshold:

$$\hat{y} = \begin{cases} 1 \text{ (backdoor)} & \text{if } \mathcal{S}_{\text{cons}}(x) > \tau_{\text{cons}} \text{ or } \mathcal{S}_{\text{var}}(x) > \tau_{\text{var}}, \\ 0 \text{ (clean)} & \text{otherwise.} \end{cases} \tag{12}$$

The use of a disjunctive criterion ensures that both trigger modalities, complete hijacking (captured by $\mathcal{S}_{\text{cons}}$) and partial hijacking (captured by $\mathcal{S}_{\text{var}}$), are independently detectable, making the method robust to diverse attack strategies. The algorithm for DetectViT is presented in Appendix A.

5 Experimental Setup

Attacked Models and Datasets. We evaluate DetectViT against four representative backdoor attacks spanning diverse ViT architectures and downstream tasks. Specifically, we consider BadViT [58] and TrojViT [61], both of which employ optimized patch-based triggers, targeting image classification models based on DeiT-T and DeiT-S [47], evaluated on Caltech-101 [25] and Tiny ImageNet [24], respectively. We further consider BadCLIP [30], which also employs an optimized patch-based trigger to attack the CLIP [42] vision-language model based ViT-B-32 [10], evaluated on ImageNet-1K [8], and BadVision [34], which employs a blended-based trigger to poison the vision encoder of LLaVA-1.5 [31] (using clip-vit-large-patch14-336), evaluated on COCO Caption [4]. We poison 5% of the test data by default. More details can be seen in Appendix B.

Table 1: Comparison of backdoor sample detection methods. TPR and FPR are reported as percentages (%). AUC is reported on a $[0, 1]$ scale. We bold the best value.

Defense	BadViT			TrojViT			BadCLIP			BadVision		
	TPR ↑	FPR ↓	AUC ↑	TPR ↑	FPR ↓	AUC ↑	TPR ↑	FPR ↓	AUC ↑	TPR ↑	FPR ↓	AUC ↑
TeCo [32]	77.95	61.86	0.63	55.00	51.76	0.51	99.20	39.52	0.89	-	-	-
Scale-up [16]	88.65	30.66	0.81	85.00	41.98	0.74	91.84	65.03	0.66	-	-	-
PatchProc [9]	90.61	18.42	0.89	87.60	21.44	0.85	86.97	26.78	0.82	-	-	-
DeDe [18]	63.97	23.29	0.78	50.20	25.80	0.72	55.12	27.97	0.69	7.00	29.68	0.50
B-IDS [21]	80.79	9.73	-	5.00	10.84	-	93.24	16.88	-	**96.00**	15.96	-
DetectViT	**100.00**	**1.58**	**1.00**	**90.80**	**5.94**	**0.95**	**100.00**	**0.17**	**1.00**	**96.00**	**3.00**	**0.99**

Baselines. We compare DetectViT against five competitive backdoor sample detection methods. TeCo [32], Scale-Up [16], and PatchProc [9] operate on the model's output label space, flagging backdoor samples by analyzing prediction consistency or sensitivity under input transformations. DeDe [18], and BackdoorIDS [21] operate in the model's embedding space, detecting anomalous feature representations induced by the trigger. Our method is orthogonal to both paradigms, operating on the attention weights without requiring either label-level queries or embedding-level access.

Evaluation Metrics. We evaluate all methods using three standard binary classification metrics: True Positive Rate (TPR), False Positive Rate (FPR), and the Area Under the ROC Curve (AUC). TPR measures the fraction of backdoor samples correctly identified, FPR measures the fraction of clean samples incorrectly flagged, and AUC provides a threshold-independent summary of overall detection performance. Higher TPR and AUC, and lower FPR, indicate better detection.

Hyperparameters. Our method introduces minimal hyperparameters. The only configurable parameters are the source and size of the OOD calibration set used for threshold estimation. Unless otherwise specified, we use iNaturalist [17] as the default OOD source with $M = 128$ calibration samples. We further conduct ablation studies on both the choice of OOD dataset and the value of M, demonstrating that our method is robust to both choices across the seven OOD datasets. All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

6 Experimental Results

Detection Performance Compared to Baselines. Table 1 presents the detection performance of all methods across four backdoor attacks. DetectViT achieves the best results across all four attacks. BackdoorIDS assigns discrete cluster labels rather than continuous scores and thus does not yield an AUC. Further, the limitations of existing baselines are systematic rather than incidental. Label-space methods (TeCo, Scale-Up, PatchProc) are fundamentally constrained to classification tasks and are entirely inapplicable to BadVision, since LLaVA-1.5 produces free-form text output rather than discrete class labels. Embedding-space methods (DeDe, BackdoorIDS) avoid this restriction but fail on TrojViT (TPRs of 50.20% and 5.00%), since its backdoor is injected into the classification head rather than the encoder, leaving poisoned embeddings indistinguishable from clean ones. DetectViT sidesteps both failure modes by operating directly on attention weights rather than model outputs or embeddings, enabling robust detection across all settings. Beyond the attacks designed specifically for vision transformers, we further evaluate five additional attacks covering self-supervised encoders (BadEncoder [23], DRUPE [45]), clean-label poisoning (Label-Consistent [48]), frequency-domain triggers (Low-Frequency [59]), and optimized global triggers (Blended-Optimized [5]), where DetectViT is the only method that maintains an AUC of at least 0.97 across all of them (Appendix C.2).

Detection Performance under Different OOD Calibration Data Sources. Table 2 reports detection performance under seven different OOD calibration sources at $M = 64$ and $M = 128$. As the calibration data only influences the detection threshold, while not influencing the AUC value, we do not report it here. Across all datasets and both sizes, TPR remains consistently high — DetectViT achieves 100% TPR on BadViT and BadCLIP regardless of the OOD source, and maintains strong TPR on TrojViT and BadVision across all settings. FPR shows somewhat greater variation across sources, which is expected given that different OOD datasets have different distributional properties

Table 2: Detection performance under different OOD calibration sources and sizes.

OOD Source	M	BadViT		TrojViT		BadCLIP		BadVision	
		TPR ↑	FPR ↓	TPR ↑	FPR ↓	TPR ↑	FPR ↓	TPR ↑	FPR ↓
iNaturalist [17]	64	100.00	2.08	93.40	13.82	100.00	1.12	96.00	2.42
	128	100.00	1.58	90.80	5.94	100.00	0.17	96.00	3.00
Food101 [2]	64	100.00	10.37	94.60	20.33	100.00	7.52	96.00	3.42
	128	100.00	9.39	92.60	11.81	100.00	8.52	96.00	2.95
Oxford-IIIT Pet [41]	64	100.00	5.48	88.60	9.04	100.00	8.90	97.00	5.16
	128	100.00	2.94	91.20	5.78	100.00	6.91	96.00	3.26
Places365 [62]	64	100.00	5.30	91.60	8.60	100.00	5.47	92.00	0.47
	128	100.00	3.75	91.20	6.47	100.00	6.34	92.00	0.32
STL-10 [6]	64	100.00	3.82	89.20	10.39	100.00	9.20	97.00	5.84
	128	100.00	2.92	89.20	9.78	100.00	6.92	97.00	5.16
Flowers102 [39]	64	100.00	5.39	92.00	7.61	100.00	7.11	98.00	9.15
	128	100.00	3.80	89.20	3.66	100.00	5.63	97.00	5.84
PASCAL VOC [11]	64	100.00	2.61	94.20	18.21	100.00	4.16	91.00	0.16
	128	100.00	1.92	93.00	13.73	100.00	2.65	91.00	0.16

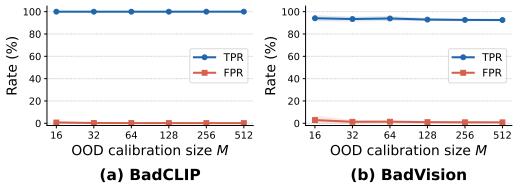


Figure 3: Effectiveness of DetectViT under different numbers of OOD calibration samples M.

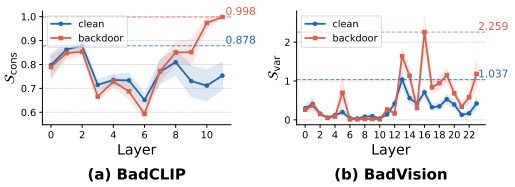


Figure 4: Visualization of detection signals across all backdoor/clean samples.

that affect threshold estimation; nevertheless, the FPR remains acceptably low for the vast majority of configurations. Notably, increasing M from 64 to 128 consistently reduces FPR across nearly all source-attack combinations, echoing the trend observed in Fig. 3. Despite the performance variation across different OOD sources, DetectViT consistently and substantially outperforms all baselines reported in Table 1. Overall, these results demonstrate that DetectViT is robust to the choice of OOD calibration source, and that practitioners can freely select from a wide range of off-the-shelf datasets without sacrificing detection reliability.

Detection Performance varying OOD Calibration Set Size. Fig. 3 illustrates how detection performance varies with the number of OOD calibration samples M. Each experiment is repeated under 10 different random seeds, where the lines represent the mean value, and the shaded regions indicate the standard deviation. We find that even with as few as $M = 16$ samples, DetectViT already achieves strong detection performance, demonstrating that the proposed threshold estimation is effective even in extremely data-scarce scenarios. Further, the variance of both TPR and FPR decreases rapidly as M increases: at $M = 16$, the shaded regions remain noticeable, whereas at $M = 32$ or 64, the variance has largely converged to a negligible level. This convergence confirms that our method requires only a small number of OOD samples to produce reliable and stable thresholds, making it practical in real-world scenarios where clean calibration data is scarce. Complete results for all attacks are provided in Appendix C.3.

Visualization of Detection Signals. Fig. 4 visualizes the per-layer detection signals averaged over all clean and backdoor samples, where the lines and shaded regions represent the mean value and the standard deviation, respectively. The peak signal value of backdoor samples substantially exceeds that of clean samples, whether measured by inter-head consistency $\mathcal{S}_{\text{cons}}$ on BadCLIP or entropy variance $\mathcal{S}_{\text{var}}$ on BadVision, confirming that the two proposed signals provide a reliable and discriminative separation between backdoor and clean inputs. Moreover, the peak signal tends to occur in the deeper

layers of the model, suggesting that backdoor features are more prominently represented in posterior transformer layers. Complete results are provided in Appendix C.4.

Runtime of Detection Methods. Table 3 reports the average per-sample detection time (in seconds) of all methods evaluated over 100 test samples on each attacked model. DetectViT achieves the lowest detection time across all four settings. Most baselines incur substantially higher latency because they rely on augmenting each test sample into multiple transformed copies before analysis. In particular, TeCo additionally applies computationally expensive image perturbations such as blurring, resulting in detection times of up to 4.29s per sample, more than 300× slower than DetectViT on TrojViT. DeDe achieves competitive per-sample inference time owing to its lightweight decoder, but this comes at the cost of a significant one-time training overhead to train the decoder. Notably, the reported runtime *upper-bounds* DetectViT's true detection cost: the bulk is the forward pass, while computing cosine similarities and entropies over the resulting attention weights is negligible by comparison. In practice, this forward pass overhead overlaps almost entirely with standard inference.

Table 3: Average per-sample detection time (s) on each attacked model. We bold the best value.

Defense	BadViT	TrojViT	BadCLIP	BadVision
TeCo [32]	2.5094	4.2920	1.8448	-
Scale-Up [16]	0.0246	0.0261	0.0696	-
PatchProc [9]	0.6686	0.6926	0.8257	-
DeDe [18]	0.0206	0.0279	0.1034	0.6060
B-IDS [21]	0.1734	0.1982	0.2560	3.3458
DetectViT	**0.0124**	**0.0134**	**0.0163**	**0.0226**

Ablation Study. Table 4 reports the contribution of each detection signal of DetectViT. For the three patch-based attacks (BadViT, TrojViT, BadCLIP), the consistency signal alone achieves near-perfect detection, confirming that complete attention hijacking, wherein all heads converge onto the trigger region, produces a strong and reliable inter-head consistency anomaly. Interestingly, for BadViT and BadCLIP, penalizing $\mathcal{S}_{\text{cons}}$ alone inadvertently induces high inter-head entropy variance, so $\mathcal{S}_{\text{var}}$ takes over, and DetectViT still achieves full detection while the attack remains near-perfect. In contrast, for the blended-based attack BadVision, the consistency signal completely fails, as the stealthy trigger is insufficiently salient to force all heads to converge onto the same region. The entropy variance signal, however, achieves 96.00% TPR with a 3.00% FPR, demonstrating that partial attention hijacking is exclusively captured by the concentration disparity across heads. The results confirm that the two signals are complementary and their combination is effective across both trigger modalities.

Table 4: Ablation study on the two detection signals. ✓indicates the corresponding signal is enabled. We bold the best value.

Signal		BadViT		TrojViT		BadCLIP		BadVision	
$\mathcal{S}_{\text{cons}}$	$\mathcal{S}_{\text{var}}$	TPR ↑	FPR ↓	TPR ↑	FPR ↓	TPR ↑	FPR ↓	TPR ↑	FPR ↓
✓		**100.00**	**0.54**	90.40	4.22	**100.00**	**0.02**	0.00	**0.00**
	✓	80.35	1.06	44.20	**1.76**	89.52	0.16	**96.00**	3.00
✓	✓	**100.00**	1.58	**90.80**	5.94	**100.00**	0.17	**96.00**	3.00

Adaptive Attacks. We evaluate DetectViT against an adaptive adversary who penalizes our detection signals during backdoor training: $\mathcal{S}_{\text{cons}}$ for patch-based attacks and $\mathcal{S}_{\text{var}}$ for blended-based attacks, or both simultaneously. Table 5 reports TPR, FPR, Attack Success Rate (ASR), and Clean Accuracy (CA), where the CA of BadVision represents the CIDEr score [50]. For BadViT and BadCLIP, penalizing $\mathcal{S}_{\text{cons}}$ alone inadvertently induces inter-head entropy variance, allowing $\mathcal{S}_{\text{var}}$ to maintain full detection with near-perfect ASR preserved. Under joint regularization, BadCLIP's ASR collapses to 14.93%, while BadViT maintains ASR at the cost of severe CA degradation (40.66%), reflecting the irreconcilable conflict between evading both signals and preserving model utility. TrojViT results are identical across all settings, since the backdoor

Table 5: Performance under adaptive attacks."w/o" denotes training without adaptive regularization.

Attack	Adaptive	TPR ↑	FPR ↓	ASR ↑	CA ↑
BadViT	w/o	100.00	1.58	100.00	83.04
	$\mathcal{S}_{\text{cons}}$	100.00	1.88	99.84	81.67
	$\mathcal{S}_{\text{cons}}$ + $\mathcal{S}_{\text{var}}$	100.00	1.32	100.00	40.66
TrojViT	w/o	90.80	5.94	100.00	85.17
	$\mathcal{S}_{\text{cons}}$	90.80	5.94	100.00	85.17
	$\mathcal{S}_{\text{cons}}$ + $\mathcal{S}_{\text{var}}$	90.80	5.94	100.00	85.17
BadCLIP	w/o	100.00	0.17	99.99	61.39
	$\mathcal{S}_{\text{cons}}$	100.00	1.95	99.96	61.06
	$\mathcal{S}_{\text{cons}}$ + $\mathcal{S}_{\text{var}}$	0.32	0.61	14.93	61.42
BadVision	w/o	96.00	3.00	99.00	85.40
	$\mathcal{S}_{\text{var}}$	0.00	0.00	35.00	85.20
	$\mathcal{S}_{\text{cons}}$ + $\mathcal{S}_{\text{var}}$	0.00	0.26	11.00	83.00

is injected into the classification head without modifying the encoder, confirming that our signals generalize to any trigger that influences attention, like adversarial examples. For BadVision, single regularization reduces ASR to 35.00%; joint regularization further degrades it to 11.00%. These results confirm that the proposed signals are unavoidable fingerprints of ViT backdoors, which cannot be suppressed without simultaneously destroying the attack.

7 Conclusion

This work probes the effect of backdoor triggers on the multi-head attention of ViTs and identifies two anomalous behaviors: complete and partial attention hijacking. Then, we propose **DetectViT**, a test-time backdoor detection method that quantifies these behaviors via an inter-head consistency score and an inter-head entropy variance score, with a small set of OOD calibration samples. Extensive experiments across various representative attacks, tasks, and models demonstrate that DetectViT outperforms all baselines, with negligible additional time overhead owing to its reliance solely on attention weights. Notably, DetectViT requires neither training data, model outputs, nor any prior knowledge of the trigger type, and applies to diverse ViT-based architectures and downstream tasks.

Limitations. While DetectViT demonstrates strong performance across diverse settings, several limitations remain. First, it cannot defend against OOD-as-trigger attacks [36], where the backdoor is activated by natural OOD features without any crafted trigger, making poisoned samples indistinguishable from clean ones at inference time. Such challenges are shared by existing test-time detection methods. Second, the detection signals rely on CLS-to-patch attention weights, which may not generalize to architectures without a CLS token, leaving extension to such settings as future work.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (Grant No. 62476095), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2025A1515011525), and the Major Science and Technology Project of Yunnan Province (Grant No. 202502AD080017). The authors declare no competing interests.

References

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[5] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[6] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[7] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International conference on machine learning*, pages 7480–7512. PMLR, 2023.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Khoa D Doan, Yingjie Lao, Peng Yang, and Ping Li. Defending backdoor attacks on vision transformer via patch processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 506–515, 2023.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[12] Shiwei Feng, Guanhong Tao, Siyuan Cheng, Guangyu Shen, Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. Detecting backdoors in pre-trained encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16352–16362, 2023.

[13] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pages 113–125, 2019.

[14] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[15] Ji Guo, Hongwei Li, Wenbo Jiang, and Guoming Lu. Backdoor attack against vision transformers via attention gradient-based image erosion. In *GLOBECOM 2024-2024 IEEE Global Communications Conference*, pages 3140–3145. IEEE, 2024.

[16] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] Grant Van Horn and macaodha. inat challenge 2021 - fgvc8. `https://kaggle.com/competitions/inaturalist-2021`, 2021. Kaggle.

[18] Sizai Hou, Songze Li, and Duanyi Yao. Dede: Detecting backdoor samples for ssl encoders via decoders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20675–20684, 2025.

[19] Siquan Huang, Yijiang Li, Chong Chen, Ying Gao, and Xiping Hu. Fedid: Enhancing federated learning security through dynamic identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(10):8907–8922, 2025.

[20] Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-metrics adaptively identifies backdoors in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4652–4662, October 2023.

[21] Siquan Huang, Yijiang Li, Ningzhi Gao, Xingfu Yan, Leyu Shi, and Ying Gao. Backdoorids: Zero-shot backdoor detection for pretrained vision encoder. *arXiv preprint arXiv:2603.11664*, 2026.

[22] Siquan Huang, Yijiang Li, Xingfu Yan, Ying Gao, Chong Chen, Leyu Shi, Biao Chen, and Wing WY Ng. Scope: On detecting constrained backdoor attacks in federated learning. *IEEE Transactions on Information Forensics and Security*, 20:3302–3315, 2025.

[23] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE, 2022.

[24] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[25] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 4 2022.

[26] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[27] Yijiang Li, Wentian Cai, Ying Gao, Chengming Li, and Xiping Hu. More than encoder: Introducing transformer decoder to upsample. In *2022 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 1597–1602. IEEE, 2022.

[28] Yijiang Li, Qingying Gao, Tianwei Zhao, Bingyang Wang, Haoran Sun, Haiyun Lyu, Robert D Hawkins, Nuno Vasconcelos, Tal Golan, Dezhi Luo, et al. Core knowledge deficits in multi-modal language models. *arXiv preprint arXiv:2410.10855*, 2024.

[29] Yijiang Li, Xinjiang Wang, Lihe Yang, Litong Feng, Wayne Zhang, and Ying Gao. Diverse cotraining makes strong semi-supervised segmentor. *arXiv preprint arXiv:2308.09281*, 2023.

[30] Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24645–24654, 2024.

[31] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[32] Xiaogeng Liu, Minghui Li, Haoyu Wang, Shengshan Hu, Dengpan Ye, Hai Jin, Libing Wu, and Chaowei Xiao. Detecting backdoors during the inference stage based on corruption robustness consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16363–16372, 2023.

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[34] Zhaoyi Liu and Huan Zhang. Stealthy backdoor attack in self-supervised learning vision encoders for large vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25060–25070, 2025.

[35] Peizhuo Lv, Hualong Ma, Jiachen Zhou, Ruigang Liang, Kai Chen, Shengzhi Zhang, and Yunfei Yang. Dbia: Data-free backdoor injection attack against transformer networks. *arXiv preprint arXiv:2111.11870*, 2021.

[36] Weimin Lyu, Jiachen Yao, Saumya Gupta, Lu Pang, Tao Sun, Lingjie Yi, Lijie Hu, Haibin Ling, and Chao Chen. Backdooring vision-language models with out-of-distribution data. *arXiv preprint arXiv:2410.01264*, 2024.

[37] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

[38] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multi-modal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022.

[39] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[40] Minzhou Pan, Yi Zeng, Lingjuan Lyu, Xue Lin, and Ruoxi Jia. Asset: Robust backdoor data detection across a multiplicity of deep learning paradigms. In *32nd USeNIX security symposium (USeNIX security 23)*, pages 2725–2742, 2023.

[41] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[43] Xuankun Rong, Wenke Huang, Jian Liang, Jinhe Bi, Xun Xiao, Yiming Li, Bo Du, and Mang Ye. Backdoor cleaning without external guidance in mllm fine-tuning. *arXiv preprint arXiv:2505.16916*, 2025.

[44] Zhengyao Song, Meixi Zheng, Ke Xu, Yongqiang Li, and Baoyuan Wu. Test-time poisoned sample detection by exploiting shallow malicious matching in backdoored clip. In *The Fourteenth International Conference on Learning Representations*, 2026.

[45] Guanhong Tao, Zhenting Wang, Shiwei Feng, Guangyu Shen, Shiqing Ma, and Xiangyu Zhang. Distribution preserving backdoor attack in self-supervised learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 2029–2047. IEEE, 2024.

[46] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021.

[48] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[51] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[52] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[53] Zhishen Wang, Rui Wang, and Lihua Jing. Attention-imperceptible backdoor attacks on vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8241–8249, 2025.

[54] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems*, 35:10546–10559, 2022.

[55] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[56] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[57] Danni Yuan, Mingda Zhang, Shaokui Wei, Li Liu, and Baoyuan Wu. Activation gradient based poisoned sample detection against backdoor attacks. In *The Thirteenth International Conference on Learning Representations*, 2025.

[58] Zenghui Yuan, Pan Zhou, Kai Zou, and Yu Cheng. You are catching my attention: Are vision transformers bad learners under backdoor attacks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24605–24615, 2023.

[59] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16453–16461. IEEE, 2021.

[60] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022.

[61] Mengxin Zheng, Qian Lou, and Lei Jiang. Trojvit: Trojan insertion in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4025–4034, 2023.

[62] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[63] Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework. *arXiv preprint arXiv:2203.14415*, 2022.

[64] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Appendix

A Main Algorithm

The full detection procedure is summarized in Algorithm 1.

Algorithm 1 DetectViT

Require: ViT encoder f, calibration set $\mathcal{D}_{\text{cal}} = \{x_m\}_{m=1}^{M}$, test sample x
Ensure: Prediction $\hat{y} \in \{0, 1\}$

// Phase 1: Threshold Estimation
for each $x_m \in \mathcal{D}_{\text{cal}}$ **do**
    Extract attention weights $\{\mathbf{A}^{(\ell,h)}\}$ from $f(x_m)$
    Compute $\tilde{\mathbf{a}}^{(\ell,h)}$ via (2) for all ℓ, h
    Compute $\mathcal{S}_{\text{cons}}(x_m)$ via (3)–(5) and $\mathcal{S}_{\text{var}}(x_m)$ via (6)–(8)
end for
Estimate $\mu_{\text{cons}}, \sigma_{\text{cons}}, \mu_{\text{var}}, \sigma_{\text{var}}$ via (9)–(10)
Set thresholds $\tau_{\text{cons}}, \tau_{\text{var}}$ via (11)
// Phase 2: Test Sample Detection
Extract attention weights $\{\mathbf{A}^{(\ell,h)}\}$ from $f(x)$
Compute $\mathcal{S}_{\text{cons}}(x)$ via (3)–(5) and $\mathcal{S}_{\text{var}}(x)$ via (6)–(8)
if $\mathcal{S}_{\text{cons}}(x) > \tau_{\text{cons}}$ **or** $\mathcal{S}_{\text{var}}(x) > \tau_{\text{var}}$ **then**
    return $\hat{y} = 1$ ▷ backdoor sample
else
    return $\hat{y} = 0$ ▷ clean sample
end if



(a) BadViT (b) TrojViT (c) BadCLIP (d) BadVision

Figure 5: Visualization of backdoor samples for each attack.

B Detailed Description of Experimental Setup

B.1 Attack Details

We describe the attacks in our experiment in detail as follows:

- **BadViT [58]:** This model fine-tunes a pretrained DeiT-T vision encoder [47] on the Caltech-101 dataset while injecting the backdoor. The trigger is an optimized patch located in the top-left corner of the image with a size the same as a patch, as shown in Fig. 5a. The original image size is 224×224. We set the target label index to 0. Other training parameters are set to match those in the original paper.
- **TrojViT [61]:** This model fine-tunes a pretrained DeiT-S vision encoder [47] on the Tiny-ImageNet dataset while injecting the backdoor into the attention-related weights of the classification head rather than the encoder. Unlike fixed-position triggers, TrojViT employs multiple optimized patches whose positions are adaptively determined for each input image based on the attention map, as shown in Fig. 5b. The original image size is 224×224. We set the target label index to 2. Other training parameters are set to match those in the original paper.
- **BadCLIP [30]:** This attack targets the CLIP [42] model (ViT-B-32 as the vision encoder) and pretrains it while injecting the backdoor on the CC3M dataset. The trigger is an optimized patch located in the center of the image with a size of 32×32, as shown in Fig. 5c. The original image size in the CC3M dataset is 224×224. The reference label for the backdoor is "banana". Other training parameters are set to match those in the original paper.
- **BadVision [34]:** This attack targets the vision encoder of LLaVA-1.5 [31] (clip-vit-large-patch14-336 as the vision backbone) and pretrains it while injecting the backdoor on the CGD subset of MIMIC-IT. The backdoor trigger is an optimized blended-based pattern with a size of 336×336, covering the entire image, as shown in Fig. 5d. The reference concept for the backdoor is "plane", and the attack is considered successful if the generated caption contains this concept, e.g., *The image features a large airplane parked on a runway, with its nose pointed towards the sky*. Other training parameters are set to match those in the original paper.

B.2 Test Data Details

We describe the test data in our experiment in detail as follows:

- **Caltech-101 [25]:** The Caltech-101 dataset consists of images belonging to 101 object categories plus a background category, with each category containing between 40 and 800 images. Images vary in size but are typically around 300×200 pixels, and are resized to $224 \times 224 \times 3$ during preprocessing. The dataset contains $9,144$ images in total, and is widely used for evaluating image classification models. After poisoning 5% of the data, the number of backdoor images is 458, and the number of clean images is $8,686$.
- **Tiny-ImageNet [24]:** The Tiny-ImageNet dataset is a subset of ImageNet, consisting of images labeled with one of 200 object categories. Each image is of size $64 \times 64 \times 3$ and

is resized to $224 \times 224 \times 3$ during preprocessing. The dataset contains $100,000$ training images and $10,000$ validation images, and is widely used for evaluating lightweight image classification models. After poisoning 5% of the data, the number of backdoor images is 500, and the number of clean images is $9,500$.

- **ImageNet 1K [8]:** The ImageNet 1K dataset consists of images labeled with one of $1,000$ object categories. Each image is of size $256 \times 256 \times 3$ and contains a single object from the corresponding category. The dataset contains 1.2 million training images and $50,000$ validation images, and is widely used for evaluating image classification models. After poisoning 5% of the data, the number of backdoor images is $2,500$, and the number of clean images is $47,500$.
- **COCO Caption [4]:** The COCO Caption dataset is part of the larger COCO dataset, designed for image captioning and other vision-related tasks. It contains over $330,000$ images, with more than $200,000$ annotated with multiple captions describing the image's content. For our experiments, we randomly select $2,000$ samples to construct the test dataset, following the methodology of [34]. After poisoning 5% of the data, the number of backdoor images is 100, and the number of clean images is $1,900$.

C Additional Experiments

C.1 Validity of the Gaussian Assumption

The three-sigma rule in Eq. (11) assumes that the detection scores of clean samples are approximately Gaussian. We examine this assumption on the clean test samples of each attacked model, using the inter-head consistency score $\mathcal{S}_{\text{cons}}$ for the three patch-based attacks (BadViT, TrojViT, BadCLIP) and the inter-head entropy variance score $\mathcal{S}_{\text{var}}$ for the blended-based attack (BadVision), consistent with Appendix C.4.

We report two complementary measures in Table 6. The first is the Shapiro-Wilk statistic W, which equals 1 for a perfectly Gaussian sample. The second is the fraction of samples falling within one, two, and three standard deviations of the mean, which we compare against the theoretical Gaussian values of 68.27%, 95.45%, and 99.73%.

Table 6: Gaussianity of the clean detection scores. We report the fraction of clean samples (%) within one, two, and three standard deviations of the mean, together with the Shapiro-Wilk statistic W. Deviations from the theoretical Gaussian values are given in parentheses.

	1-sigma	2-sigma	3-sigma	Shapiro-Wilk W
Theoretical	68.27	95.45	99.73	1.0000
BadViT	69.80 (+1.53)	94.65 (−0.80)	99.79 (+0.06)	0.9890 (−0.0110)
TrojViT	77.71 (+9.44)	94.66 (−0.79)	99.09 (−0.64)	0.9179 (−0.0821)
BadCLIP	69.06 (+0.79)	95.21 (−0.24)	99.61 (−0.12)	0.9973 (−0.0027)
BadVision	67.47 (−0.80)	95.32 (−0.13)	99.68 (−0.05)	0.9968 (−0.0032)

On BadViT, BadCLIP, and BadVision, the clean scores are close to Gaussian: W is near 1, and the coverage at all three levels deviates from theory by less than 1.6 percentage points. TrojViT departs the most ($W = 0.9179$), but the departure is concentrated at the peak rather than in the tails: 77.71% of its clean scores fall within one standard deviation against a theoretical 68.27%, meaning that its distribution is more sharply concentrated around the mean than a Gaussian, which favors detection. Its two- and three-sigma coverage remains close to theory, so the tail behavior on which the three-sigma threshold depends is unaffected.

To examine whether the detection performance hinges on the Gaussian assumption itself, we replace the three-sigma threshold with a distribution-free alternative, namely the 99th percentile of the OOD calibration scores, keeping all other settings unchanged. As the threshold does not affect the AUC, we report only TPR and FPR in Table 7.

The two thresholds yield comparable detection performance across all four attacks, indicating that the effectiveness of DetectViT stems from the separability of the two signals rather than from the

Table 7: Detection performance under the three-sigma threshold and the 99th-percentile threshold. TPR and FPR are reported as percentages (%). We bold the best value.

Threshold	BadViT		TrojViT		BadCLIP		BadVision	
	TPR ↑	FPR ↓	TPR ↑	FPR ↓	TPR ↑	FPR ↓	TPR ↑	FPR ↓
Three-sigma	**100.00**	**1.58**	**90.80**	5.94	**100.00**	**0.17**	96.00	**3.00**
99th percentile	**100.00**	4.29	89.40	**5.14**	**100.00**	0.52	**97.00**	6.32

particular estimator used to derive the threshold. The three-sigma rule nonetheless achieves a lower FPR on three of the four attacks, which supports its use as the default.

C.2 Evaluation on Additional Attacks

To examine whether the two detection signals generalize beyond the attacks in the main evaluation, we consider five additional attacks spanning different threat models and trigger types: BadEncoder [23] and DRUPE [45], which implant backdoors into self-supervised visual encoders; Label-Consistent [48], a clean-label attack; Low-Frequency [59], which places the trigger in the frequency domain; and Blended-Optimized, which replaces the fixed pattern of the blended attack [5] with an optimized global trigger.

For BadEncoder and DRUPE, we follow the ViT protocol specified in the appendix of DRUPE [45]: we use Mugs [63] as the self-supervised learning algorithm, pretrain ViT-T on CIFAR-10, and use GTSRB as the downstream task. For Label-Consistent and Low-Frequency, we reproduce the attacks from the implementations collected in BackdoorBench [54], training and testing ViT-S on Tiny-ImageNet. For Blended-Optimized, we replace the "kitty" pattern with an optimized global trigger and train within the BadCLIP framework (CLIP ViT-B/32), evaluating on ImageNet-1K. The detection settings (iNaturalist, $M = 128$) and the baseline implementations are identical to those in the main evaluation.

Table 8: Detection performance on five additional attacks. TPR and FPR are reported as percentages (%). AUC is reported on a $[0, 1]$ scale. We bold the best value.

Defense	BadEncoder			DRUPE			Low-Frequency			Label-Consistent			Blended-Optimized		
	TPR ↑	FPR ↓	AUC ↑	TPR ↑	FPR ↓	AUC ↑	TPR ↑	FPR ↓	AUC ↑	TPR ↑	FPR ↓	AUC ↑	TPR ↑	FPR ↓	AUC ↑
TeCo	54.91	62.29	0.37	48.10	65.79	0.35	99.60	62.38	0.52	66.80	56.99	0.57	12.12	29.67	0.42
Scale-up	63.29	10.26	0.79	47.15	6.93	0.72	**100.00**	11.32	0.95	84.00	13.06	0.88	95.24	17.39	0.91
PatchProc	85.44	21.58	0.83	75.95	23.68	0.78	**100.00**	17.05	0.87	0.00	20.21	0.47	88.20	20.86	0.85
DeDe	99.21	23.13	0.97	**97.94**	27.37	0.91	0.00	30.33	0.00	78.60	25.52	0.83	73.44	23.98	0.81
B-IDS	96.04	17.12	-	95.09	22.11	-	0.00	26.00	-	14.60	25.47	-	44.96	12.78	-
DetectViT	**99.37**	**5.33**	**0.99**	95.57	**4.42**	**0.99**	93.40	**7.11**	**0.97**	**94.00**	**6.52**	**0.97**	**99.08**	**0.64**	**1.00**

As shown in Table 8, DetectViT attains the best value in most cells and is the only method that maintains an AUC of at least 0.97 across all five attacks, whereas each baseline degrades severely on at least one of them. Label-based methods are sensitive to the trigger type: PatchProc fails to detect Label-Consistent (TPR 0.00%), and TeCo reaches an AUC of only 0.42 on Blended-Optimized. Embedding-based methods fail on Low-Frequency, where both DeDe and BackdoorIDS detect none of the backdoor samples, suggesting that frequency-domain triggers leave the encoder representation largely indistinguishable from that of clean inputs.

DRUPE provides a notable case. Although it uses a patch trigger, its distribution-preserving objective suppresses inter-head consistency in pursuit of stealth and raises the inter-head entropy variance instead, so it is detected almost entirely by $\mathcal{S}_{\text{var}}$ rather than $\mathcal{S}_{\text{cons}}$. This mirrors the behavior observed under adaptive attacks in Table 5, where suppressing one signal induces the other. It also indicates that the visual form of a trigger does not reliably determine which signal is triggered; the two signals are complementary, and it is their disjunctive combination that makes the detection robust across trigger types.

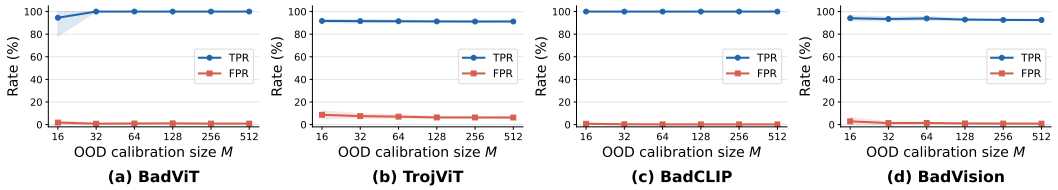


Figure 6: Effectiveness of DetectViT under different numbers of OOD calibration samples M.

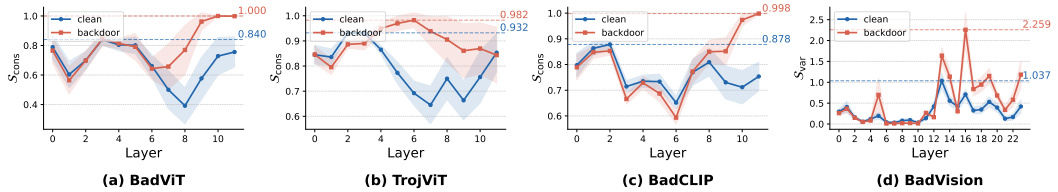


Figure 7: Mean value of detection signals across all backdoor/clean samples.

C.3 Detection Performance varying OOD Calibration Set Size

Fig. 6 illustrates how detection performance varies with the number of OOD calibration samples M across all four attacks. Each experiment is repeated under 10 different random seeds, where the solid lines represent the mean value and the shaded regions indicate the standard deviation. Two consistent observations emerge. First, even with as few as $M = 16$ samples, DetectViT already achieves strong detection performance across all attacks, demonstrating that the proposed threshold estimation is effective even in extremely data-scarce scenarios. Second, the standard deviation shrinks rapidly as M increases: at $M = 16$, the shaded region is noticeable on BadViT and BadVision, whereas at $M = 32$ or 64, the variance is negligible across all attacks, indicating that the thresholds have largely stabilized. This rapid convergence confirms that our method requires only a small number of OOD samples to produce reliable and stable thresholds. Based on these results, we adopt $M = 128$ as the default setting, which provides a favorable balance between stability and calibration cost.

C.4 Visualization of Detection Signals

Fig. 7 visualizes the per-layer detection signals averaged over all clean and backdoor samples across four attacks, where the solid lines and shaded regions represent the mean and standard deviation, respectively. For the three patch-based attacks (BadViT, TrojViT, BadCLIP), we report the inter-head cosine consistency $\mathcal{S}_{\text{cons}}$; for the blended-based attack BadVision, we report the inter-head entropy variance $\mathcal{S}_{\text{var}}$. In all cases, the peak signal value of backdoor samples substantially exceeds that of clean samples — 1.000 vs. 0.840 on BadViT, 0.982 vs. 0.932 on TrojViT, 0.998 vs. 0.878 on BadCLIP, and 2.259 vs. 1.037 on BadVision — confirming that the two proposed signals provide a reliable and discriminative separation between backdoor and clean inputs. Moreover, the peak signal tends to occur in the deeper layers of the model, suggesting that backdoor features are more prominently represented in posterior transformer layers.

D Code Availability

We provide the implementation and detailed guidelines to reproduce our work in the code repository: `https://github.com/siquanhuang/DetectViT`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss the limitations of our work in the section of conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper is empirical in nature and does not include formal theorems or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 5 and Appendix B provide detailed descriptions of the algorithm, attacked models, datasets, baselines, hyperparameters, and evaluation metrics. The full algorithm is given in Appendix A, and code is released at `https://github.com/siquanhuang/DetectViT`.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We release code with reproduction instructions at `https://github.com/siquanhuang/DetectViT` (Sec. 5 and Appendix D). All datasets used (Caltech-101, Tiny-ImageNet, ImageNet-1K, COCO Caption, and the OOD calibration datasets) are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5 specifies attacked models, datasets, baselines, evaluation metrics, hyperparameters (default OOD source iNaturalist with M=128), and hardware (single NVIDIA RTX 4090). Appendix B provides further details on attacks, trigger configurations, target labels, and test data splits.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For experiments with random calibration sampling (Fig. 3 and Fig. 6), each setting is repeated under 10 random seeds, and we report the mean values along with shaded regions indicating the standard deviation. Per-layer detection signal visualizations (Fig. 4 and Fig. 7) likewise report mean and standard deviation across all samples.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 5 states that all experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU. Per-sample detection runtime for all methods is reported in Table 3, demonstrating that DetectViT introduces negligible overhead since it reuses attention weights already computed during inference.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our work has clear positive societal impact: it provides a practical defense against backdoor attacks on widely deployed Vision Transformers and multimodal models, helping practitioners safely use third-party pretrained checkpoints. As a defensive method, it does not introduce direct negative societal impacts; while a sophisticated adversary could in principle adapt to our signals, our adaptive-attack experiments (Table 5) show that evasion comes at the cost of either attack success or model utility.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This paper releases a backdoor detection method rather than pretrained generative models, scraped datasets, or other assets with high misuse risk. The released code is purely defensive and does not facilitate the creation of new attacks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing assets used in this work — including DeiT, CLIP, LLaVA-1.5, and the datasets (Caltech-101, Tiny-ImageNet, ImageNet-1K, COCO Caption, iNaturalist, Food101, Oxford-IIIT Pet, Places365, STL-10, Flowers102, PASCAL VOC) — are properly cited in the references, and we use them in accordance with their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the implementation of DetectViT as a code repository (`https://github.com/siquanhuang/DetectViT`) with documentation and instructions for reproducing all reported results, as described in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing experiments or research with human subjects. All experiments are conducted on publicly available image and image-caption datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research and therefore does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only used for language polishing and editing of the manuscript.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.