

A Classifier-Agnostic Zero-Shot Adversarial Attack Detection via CLIP

Hodaya Krakover¹, Meir Yossef Levi¹, Eyal Gofer¹, and Guy Gilboa^{1*}

Technion - Israel Institute of Technology, Haifa, Israel
hodaya.r@campus.technion.ac.il

Abstract. Adversarial attacks pose a challenge to the reliability of deep learning models, motivating effective detection methods. Existing techniques often rely on attack-specific assumptions, access to adversarial samples, or knowledge of the underlying classifier (white-box). We propose A^4D (*Attack- and Architecture-Agnostic Adversarial Detector*), a completely black-box, zero-shot adversarial attack detection framework that utilizes prompt-based similarity scores derived from CLIP. To the best of our knowledge this is the first attempt to utilize CLIP for such a task. The method is based on two key observations: (i) CLIP is sensitive even to small imperceptible non-semantic perturbations; (ii) The shift in CLIP embedding space is not arbitrary and can be used as a robust attack indicator. Experiments across multiple attacks, datasets and classifiers validate that A^4D achieves SOTA detection results in the attack-agnostic and classifier-agnostic setting.

Keywords: Adversarial Attack · Zero-shot Detection · Classifier Agnostic

1 Introduction

One of the major concerns regarding the reliability of vision encoders is that they can be fooled, sometimes easily, by adversarial attacks [6, 10, 15, 24]. While the human eye is largely agnostic to small changes in a scene, even the most advanced vision encoders tend to alter their predictions when small perturbations occur [28]. With the rise of adversarial attacks, a complementary line of research has emerged, focusing on defenses that aim to enhance the robustness of encoders to such targeted perturbations [5, 16, 18, 19, 35]. However, given the rapid progress in visual encoders, it is crucial to develop defenses that are generalizable and applicable to a wide range of models, including future ones. Moreover, since there is a wide variety of adversarial attacks, it is desirable for a defense method to remain robust regardless of the specific attack mechanism. In this paper, we focus on a zero-shot *detection* method: given only an input image (and a mini-validation set from the dataset), we aim to determine whether it has been adversarially manipulated or not.

* This work was jointly supervised by the last two authors.

Existing adversarial detection methods typically require additional information. Some approaches rely on attack-specific characteristics and therefore do not generalize well across different attack types [9, 26, 42]. Others depend on classifier-specific representations [8, 23, 30, 36, 44]. However, such information may not be available in real-world scenarios. Ideally, we would like to detect attacks without access to the underlying model or the attack mechanism, especially as new encoders and adversarial attacks continuously emerge. In parallel, a large body of work has focused on leveraging foundation models such as CLIP [32] for semantic tasks, including detection [45], segmentation [31], classification [39], and text-to-image generation [7]. These models are typically viewed as operating in a semantic latent space, designed for image-text alignment. However, this perspective is incomplete: beyond semantics, CLIP embeddings can also be exploited for non-semantic, low-level vision tasks. In particular, we show that this characteristic can be used for adversarial detection.

Our motivation is based on two key observations regarding the behavior of CLIP under adversarial perturbations. First, although adversarial changes are often visually imperceptible, CLIP embeddings are sensitive to such perturbations (previous works on ImageNet-C [12] shows that CLIP norms gradually grow as noise increases [2, 17]). Even small attacks induce a measurable shift in the representation, reflected by a non-negligible change in cosine similarity between clean and attacked images (see Fig. 1a). Second, this shift is not arbitrary and yields a distinct bias, which can be well measured (see Fig. 1b, Fig. 1c and Fig. 1d). As attacked samples concentrate in a specific region of the latent space, we can leverage the semantic structure of CLIP to characterize this region using text prompts, and identify adversarial inputs based on their similarity to it.

Based on these observations, we propose A^4D (*Attack- and Architecture-Agnostic Adversarial Detector*), a zero-shot method that operates in the embedding space of CLIP. By computing the cosine similarity between an image and a small set of dedicated captions, A^4D can detect adversarial inputs without access to the classifier or knowledge of the attack.

Our main contributions are:

- We propose to utilize CLIP as a **zero-shot adversarial detection framework**, based on the similarity to a predefined set of prompts.
- The method is **attack- and classifier-agnostic**, with limited validation-based design.
- We demonstrate the effectiveness of A^4D on multiple attacks (FGSM, PGD, AutoAttack, BIM, CW, DeepFool and Square), two Datasets (Tiny-ImageNet, StreetSurfaceVis) and four classifiers (DeiT-Small, Wide-ResNet, ResNet34 and convnext-tiny), yielding SOTA performance.
- The framework is **computationally efficient**: training requires a single forward pass to compute embeddings, and detection at test time involves lightweight operations on low-dimensional similarity vectors.

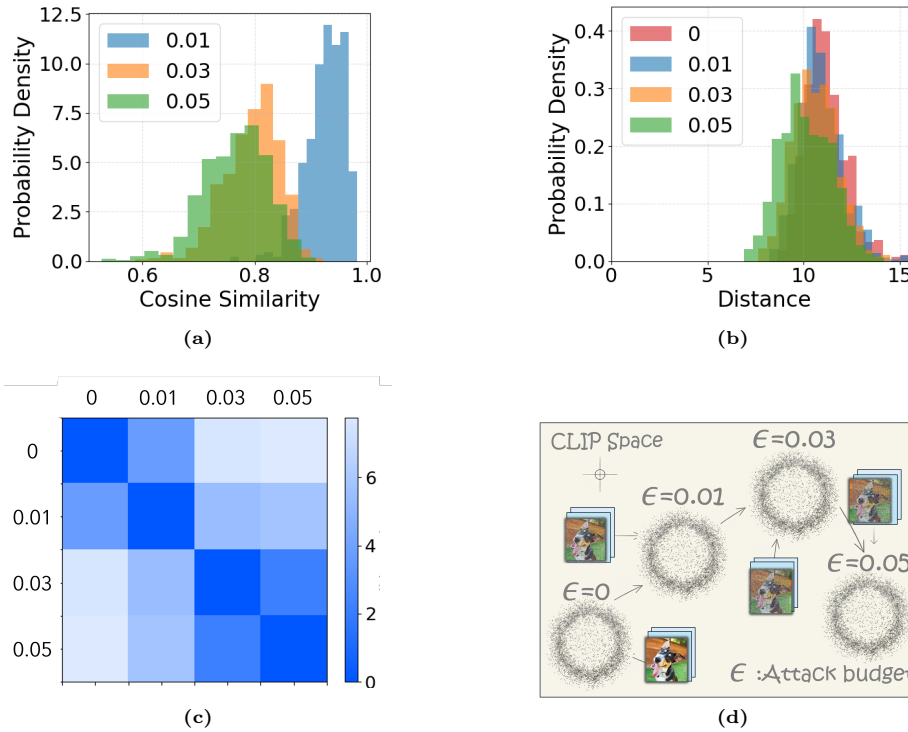


Fig. 1: CLIP sensitivity to adversarial perturbations. (a) Cosine similarity histogram between Tiny-ImageNet images and their FGSM adversarial counterparts, showing a clear embedding shift even for small budgets. (b) **Thin-shell structure.** Histogram of embedding norms with respect to the dataset mean reveals that samples lie on a thin shell [17] (much larger bias than variance). (c) **Gradual latent shift.** As the attack budget increases, the ℓ_2 distance between distribution centers grows consistently. (d) **Embedding evolution under attack.** Increasing perturbation strength progressively shifts samples in the latent space.

2 Related Work

2.1 Detection of Adversarial Attacks

Broadly, adversarial detection methods can be grouped into three categories:

1. Attack-specific detection methods. These methods are designed to detect a particular attack or a limited family of attacks. Typical algorithms include learning a threshold or training a classifier to distinguish between clean and adversarial samples using confidence scores, input statistics, or output distributions [9, 26, 42]. While this direction can be effective under the assumed threat model, it often relies on attack-dependent parameters. This implicitly assumes prior knowledge of the attack type, and in some cases even of the attack param-

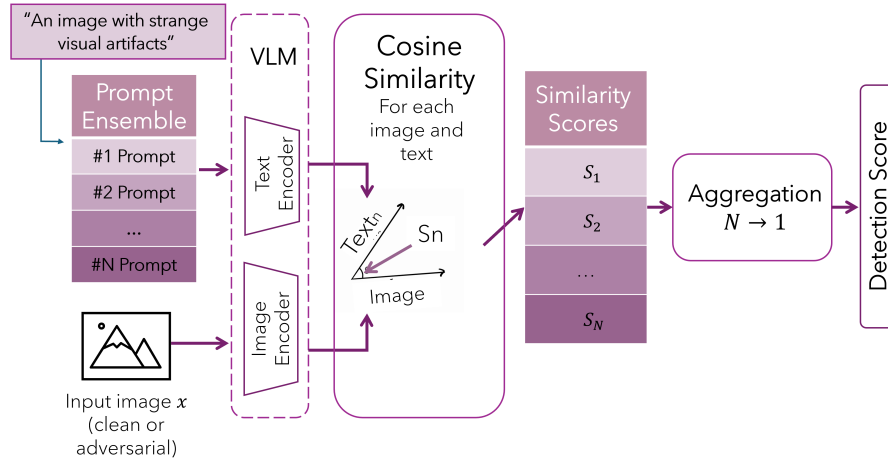


Fig. 2: Test-time zero-shot adversarial detection pipeline. Given an input image, cosine similarities between the image embedding and a fixed set of textual prompt embeddings are computed using CLIP. The resulting similarity scores are aggregated into a single detection score, where higher values indicate a higher likelihood of adversarial perturbation.

eters. This type of information is often not available for defense in real-world scenarios.

2. Internal classifier representations. These techniques use feature activations or hidden layers, to detect adversarial inputs [23,30], or exploit architecture-specific characteristics (e.g., properties of convolutional neural networks) [29]. An inherent underlying assumption is the use of a particular model structure, which may not hold for all classifiers. In these approaches, the detection algorithm is tightly coupled to the specific classifier architecture.

3. Training-based detection. These approaches rely on training a new model, either an explicit detector or a modified version of the classifier itself. In these methods, detection is often formulated as a supervised learning problem, where the model is trained using both clean and adversarial examples, e.g., [25,26]. Such approaches usually require access to attacked data during training, and their performance strongly depends on the attacks used to generate this data.

In contrast, we focus on a zero-shot detection setting that is agnostic to both the attack type and the underlying classifier. The proposed formulation does not require labeled adversarial data to characterize the distinction between clean and adversarial inputs. To the best of our knowledge, this highly challenging setting has not been addressed in prior work.

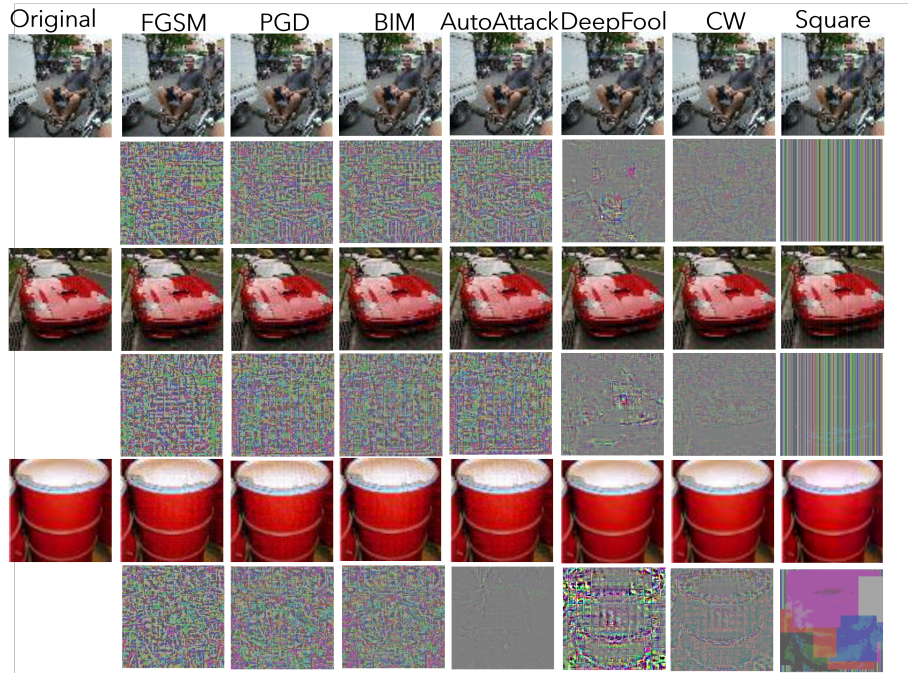


Fig. 3: Qualitative examples of clean and adversarial images under different attack methods. Three pairs of clean and adversarial images are shown. The second row in each example visualizes the difference between the adversarial and original images (with scale). All attacks are applied on the DeiT-Small classifier.

2.2 Usages of CLIP

CLIP [32] is a pioneering multi-modal encoder trained to align images and text into a shared embedding space using contrastive learning on large-scale image-text pairs. Beyond classification, CLIP has been applied to a variety of tasks, including retrieval [21, 37], open-vocabulary recognition [40, 41], and text2image [33, 34]. More recently, CLIP-based representations have been explored for out-of-distribution and anomaly detection in a zero-shot setting. For example, AnomalyCLIP [45] uses textual prompts to detect anomalous regions without task-specific training, AdaCLIP [3] adapts CLIP representations for anomaly detection across domains, and AA-CLIP [22] leverages CLIP embeddings to identify abnormal samples without access to anomaly labels. In this work we show that CLIP can be highly instrumental also in the adversarial attack detection field.

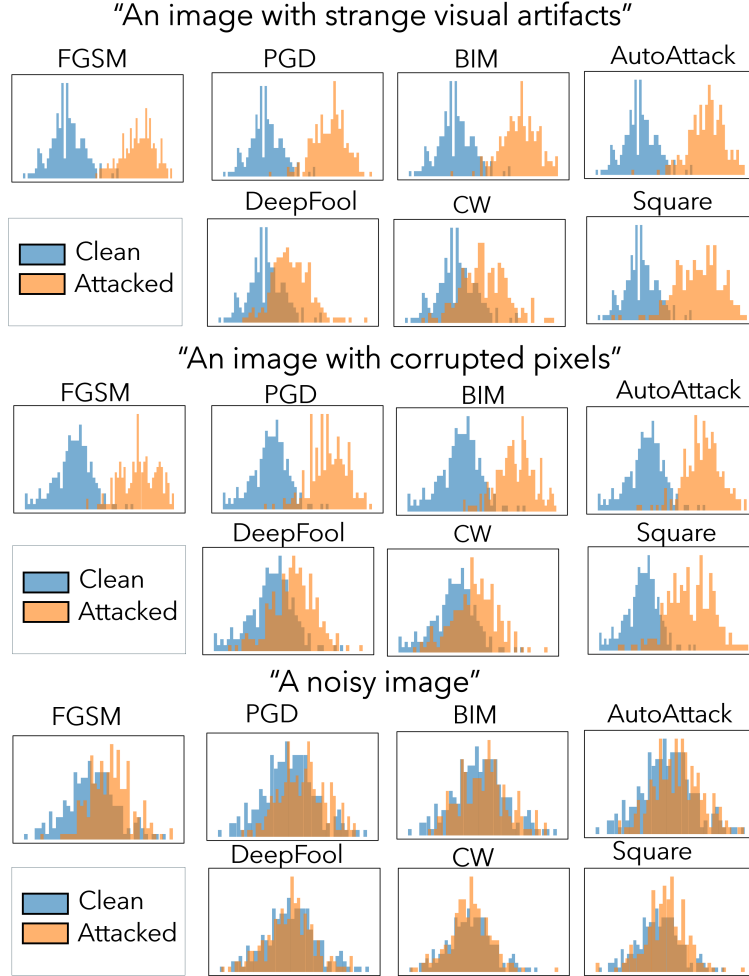


Fig. 4: Prompt-specific detection behavior. Three examples illustrating the similarity between images and three specific text prompts are shown. Only the first two prompts are included in the final prompt dictionary. For the first two prompts, a clear separation between clean and adversarial samples is observed for most attack types, with DeepFool and CW being the most challenging to distinguish. The behavior varies across prompts, as each prompt exhibits different sensitivity patterns to specific attacks. Since our goal is to maintain a zero-shot setting with the potential to generalize to previously unseen attacks, we employ multiple prompts in the final method to enhance robustness and diversity. The third prompt serves as a counterexample. Although it is semantically similar to the first two, it yields poor separation when used as a caption, highlighting the importance of careful prompt selection.

Table 1: Naïve detector. Zero-shot adversarial detection performance by measuring the distance to trajectory endpoint (AU-ROC) on Tiny-ImageNet dataset.

Method	FGSM	PGD	AutoAttack	BIM	CW	DeepFool	Square
Distance to endpoint	93.87	89.21	87.79	82.37	66.31	66.78	85.36

3 Method

3.1 Preliminaries and problem definition

Let $X \in \mathbb{R}^{H \times W \times C}$ denote an image and let $t \in \mathcal{T}$ denote a text prompt. The corresponding CLIP embeddings in the shared embedding space are denoted by $z_I \in \mathbb{R}^d$ for an image and $z_T \in \mathbb{R}^d$ for a text. The cosine similarity between the image and text embeddings is denoted by $\text{sim}(z_I, z_T) = \frac{\langle z_I, z_T \rangle}{\|z_I\| \|z_T\|}$. We denote vectors by lowercase letters and matrices by uppercase letters. A classifier $c : \mathbb{R}^{H \times W \times C} \rightarrow \{1, \dots, K\}$ maps an input image to one of K class labels. An adversarial example $X^{adv} = X + \delta$ is generated using an untargeted attack, where the perturbation is limited to a budget ϵ , such that $\|\delta\|_p \leq \epsilon$, with p corresponding to the standard norm for each attack (e.g., ℓ_∞ for FGSM, PGD, BIM, Square; ℓ_2 for DeepFool and CW), and aims to cause misclassification, i.e., $c(X^{adv}) \neq c(X)$. A detector $d : \mathbb{R}^{H \times W \times C} \rightarrow \{0, 1\}$ is tasked with correctly predicting whether an input image X' is clean or adversarial, that is, $d(X') = 0$ if X' is clean and $d(X') = 1$ if X' is adversarial.

3.2 Exploiting structured latent behavior for adversarial detection

As shown in Fig. 1d, CLIP embeddings are sensitive even to small, imperceptible perturbations in the image domain, such that $z_I^{adv} \neq z_I^{clean}$, and embedding distributions z_I^{adv} and z_I^{clean} shift as the attack budget grows. This raises the question: *How can we leverage this latent behavior to detect malicious samples?* A simple, naïve approach would be to classify images based on their distance from the expected endpoint of this trajectory (full details are in the Supp.). As shown in Table 1, while this detector achieves reasonable performance, it remains limited due to several reasons: (1) it assumes that all attacks produce similar shifts, whereas we aim for an attack-agnostic detector; (2) it ignores the semantic knowledge of CLIP, since it does not involve the text domain, which could provide a more interpretable solution. To address these limitations, we propose projecting images onto a dictionary of related captions. This approach leverages CLIP’s semantic space, reduces overfitting to specific attacks, and enhances robustness to a variety of attack mechanisms while providing a more explainable detection signal.

3.3 Our adversarial attack detection framework (A^4D)

First, we construct a dictionary of text prompts that describe the semantic changes induced by noise or adversarial attacks. Leveraging CLIP’s strong se-

Table 2: The dictionary of text prompts used for detecting adversarial images ($N = 10$).

ID	Prompt
1	“An image with noise”
2	“An image with unnatural texture”
3	“An image with digital distortions”
4	“An image with corrupted pixels”
5	“An image with adversarial perturbations”
6	“An image that was modified artificially”
7	“An image with strange visual artifacts”
8	“An image with high-frequency noise”
9	“An image that contains tampering”
10	“An image with pixel-level manipulation”

mantic understanding, we select N text prompts that capture different forms of perturbations, shown in Table 2. We employ an ensemble of prompts for two main reasons: (i) to capture variations across multiple attack types, since the embedding-space behavior of different attacks may not be identical (see Fig. 4), and (ii) to reduce variance in the detection signal, thereby stabilizing the algorithm (see the low correlation between prompts in Fig. 5).

Second, we empirically estimate the distribution of cosine similarities between clean images and the text prompt dictionary. To do this, we construct a representative set of M clean images from the dataset of interest. For each image X_i and each prompt t_j , $j = 1, \dots, N$, we compute the similarity $\text{sim}(z_I^i, z_T^j)$. This allows us to characterize the “similarity space” spanned by clean images, an N -dimensional space where each axis corresponds to the similarity with a specific prompt. Later, this space is used to determine whether a new image X' lies within the expected distribution of clean images or originates from a different distribution (e.g., adversarially perturbed).

Finally, we should aggregate the N -dimensional similarity vector to assign a single score, $s \in \mathbb{R}$, to each image. We compare several reasonable aggregation methods (see Table 4) and find that the PCA-based aggregation performs best. Therefore, we describe this method in greater detail in the next section.

3.4 Detecting adversarial attacks at inference

Let $Z_I \in \mathbb{R}^{M \times d}$ be the embedding matrix of the representative set, where M is the number of clean images and d is the embedding dimension. This representative set serves as a small validation set used solely for estimating the aggregation direction and does not require model training or fine-tuning. In practice, we use only 200 clean images sampled from the target dataset, which is sufficient to obtain a stable estimate of the principal component. Let $Z_T \in \mathbb{R}^{N \times d}$ be the embedding matrix of the dictionary of text prompts, where N is the number of prompts. We define $P \in \mathbb{R}^{M \times N}$ such that each entry $P_{ij} = \text{sim}(z_I^{(i)}, z_T^{(j)})$.

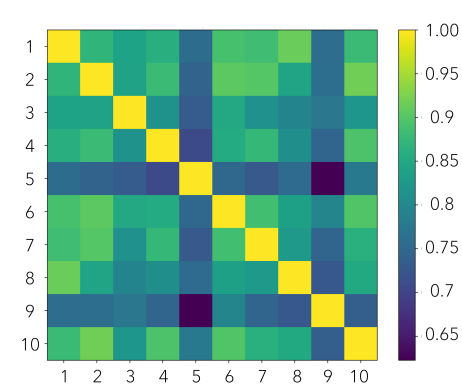


Fig. 5: Pairwise correlation heatmap of the prompt-based similarity scores computed using CLIP. Prompts are indexed according to Table 2. The low inter-prompt correlations suggest that the selected prompts provide diverse indicators.

The matrix P undergoes standardization, yielding $\tilde{P}$. Namely, each column of $\tilde{P}$ has mean 0 and standard deviation 1. Then, PCA is applied to $\tilde{P}$, and v_1 , the direction of the first principal component, is obtained.

Given a new image $X' \in \mathbb{R}^{H \times W \times C}$, we obtain its embedding z'_I and compute its similarity to dictionary prompt embeddings $u'_j = \text{sim}(z'_I, z_T^{(j)})$, $j = 1, \dots, N$, yielding $u' \in \mathbb{R}^N$. The previously calculated standardization transformation is applied to u' , yielding $\tilde{u}'$. The detection score is obtained by projecting $\tilde{u}'$ onto the first principal direction $v_1 \in \mathbb{R}^N$:

$$s' = \tilde{u}'^\top v_1. \quad (1)$$

The use of the direction of PC1 for aggregation is motivated by the fact that it is the one-dimensional linear projection that explains the maximum variance in the similarities across instances in the representative set.

The score s' is used to compute the AUC on the test set. The overall pipeline of the method is illustrated in Fig. 2.

4 Experiments

4.1 Datasets

We evaluated our method on (1) Tiny-ImageNet [27] contains 200 classes. In addition to this benchmark we validate the results on a smaller dataset (2) StreetSurfaceVis [13], which contains 5 classes of different road materials. The non-standard dataset was used as additional verification of our method since it was published after the introduction of CLIP. The second dataset allows us to validate that the results are not linked to any possible exposure of CLIP during training with the standard dataset (which was published earlier).

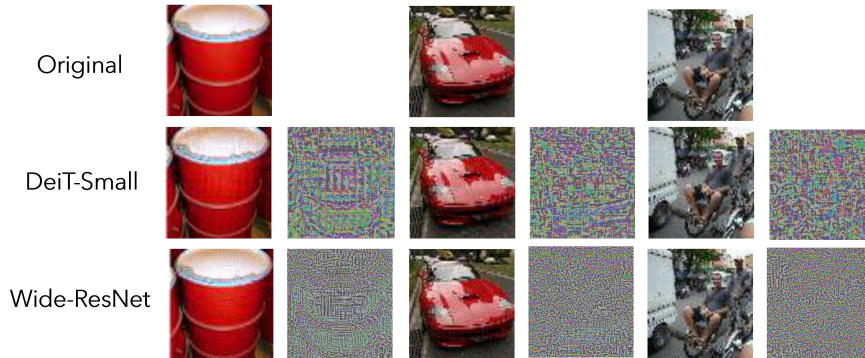


Fig. 6: Qualitative examples of clean and adversarial images under the same attack (FGSM) across different models. Three pairs of clean and adversarial images are shown. The image to the right of each clean image shows the difference between the attacked image and the clean image (with increased contrast). The second row corresponds to the DeiT-Small architecture, and the third row corresponds to the Wide-ResNet architecture. The difference images clearly show that the same attack (FGSM in this example) produces different perturbation patterns for the two classifiers.

4.2 Adversarial Attacks

We used torchattacks [14] to implement all examined attacks; They are mostly white-box: Fast Gradient Sign Method (FGSM) [10], Projected Gradient Descent (PGD) [24], Basic Iterative Method (BIM) [15], AutoAttack [6], DeepFool [28], Carlini-Wagner (CW) [4] and one black-box attack - Square Attack [1]. Examples of the attacks are shown in Fig. 3. Pay attention that while some attacks are noise-like, there exists attacks which their additive signal is more structured.

For all attacks that require a perturbation budget (FGSM, PGD, BIM, AutoAttack, and Square), we set the budget to $\epsilon = 8/255$. For iterative attacks (PGD and BIM), we use a fixed number of iterations with step size ($\alpha_{PGD} = \epsilon/4$, $\alpha_{BIM} = \epsilon/4$), chosen according to standard practice. AutoAttack and Square are evaluated using their official implementations with default parameters, except for the perturbation budget $\epsilon = 8/255$. All remaining attacks are configured following their original papers based on their official implementations.

4.3 Experimental Setup

We evaluate the proposed method on two image datasets and four classifier architectures: Wide-ResNet [43], ResNet34 [11], DeiT-Small [38] and ConvNext-Tiny [20]. The use of multiple classifiers demonstrates the classifier-agnostic nature of the method. Since each classifier responds differently to adversarial optimization, the resulting perturbations produced by the same attack may vary across models, as illustrated in Figure 6.

Runtime. Since our method is almost training-free (requires only mini validation set), we only report inference time. All experiments are conducted on a single NVIDIA RTX 3090 GPU with batch size 64. The average inference time is 0.0091 seconds per image, measured over 200 images and excluding data loading. The reported runtime includes evaluation over all textual prompts used by our method. This runtime enables real-time processing and large-scale evaluation.

4.4 Baselines

For comparison, we implement two classical classifier-dependent baselines based on noise statistics: Median Absolute Deviation (MAD) and Wavelet-based noise estimation. In addition, we re-implement a prior method for distinguishing clean and adversarial examples based on Mahalanobis [16], which was originally proposed as a generalized detection approach. To assess this claim, we evaluate two variants of this method: (i) an *attack-aware* version that is tuned separately for each attack, and (ii) an *attack-agnostic* version that does not assume knowledge of the attack type.

Due to the high computational cost of the baseline methods, these comparisons are not performed on AutoAttack of two classifier architectures. This setting follows common practice and enables a fair comparison under feasible computational constraints.

One of the evaluated classifier architectures (ResNet34) exhibits lower classification accuracy compared to the others on the Tiny-ImageNet dataset. The Mahalanobis baseline was crafted for a set of architectures including ResNet34; therefore, to enable a faithful comparison with the Mahalanobis baseline, this model was included. Following the original experimental setup allows us to minimize deviations from prior work and isolate differences which can be attributed to the detection method rather than to architectural choices.

5 Evaluation

5.1 Results

The main results of the proposed method across all datasets and classifier architectures, along with a detailed comparison with baseline methods is presented in Table 3, where results corresponding to the attack-agnostic variant of the Mahalanobis method are marked with (*). Beside on ConvNext-Tiny, where our approach performs only second-best on average, on all other architectures our method is optimal or second best on all attacks, and best on their average.

5.2 Ablation Study

To justify the choice of PCA based transformation, we evaluate its effectiveness against alternative pooling strategies. Additional common aggregation functions include min, max, and mean. To select the most effective aggregation function, we

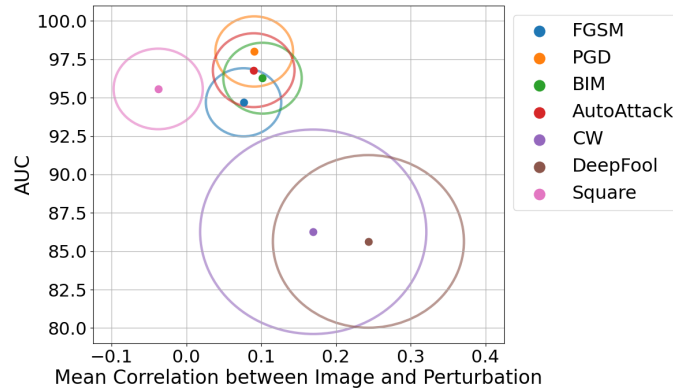


Fig. 7: Relationship between perturbation characteristics and detection performance. Each point represents a different adversarial attack. The x-axis denotes the mean correlation between the original image and the corresponding difference image (original minus adversarial), measuring the extent to which the perturbation preserves image structure. The y-axis shows the AUC of our detection method (tiny-ImageNet Dataset; DeiT-Small classifier). The radius of each circle corresponds to the standard-deviation of the correlations.

construct a validation set consisting of clean images from the dataset of interest and adversarial images generated by various attacks on a specific classifier. As reported in Table 4, PCA-based aggregation yields the best separation between clean and adversarial samples averaged on all attacks, and is therefore used in our method.

5.3 Discussion

The results are significantly better for FGSM, PGD, AutoAttack, and BIM, while the AUC decreases for CW and DeepFool, and we aim to explain this phenomenon. As we have shown, CLIP is sensitive to added noise to the image, so we check how similar the perturbation of each attack is to added white Gaussian noise. Fig. 3 qualitatively shows that the perturbations of CW and DeepFool are similar to the edges of the images, while the perturbations of the other attacks resemble noise. For a quantitative measurement, we calculate the correlation between the edges in the perturbations and the edges of the images. The edges were calculated using the Sobel algorithm. The results are shown in Fig 7. The x-axis represents the correlation value, and the y-axis represents the AUC of our method on one dataset and one classifier (Tiny-ImageNet; DeiT-Small). For each attack, there is a circle—the center represents the mean of 200 correlations, and the radius represents the standard deviation. CW and DeepFool perturbations have a higher correlation with the image edges, as expected.

Table 3: Zero-shot adversarial detection performance (AU-ROC) for multiple architectures and attack methods. The competitors (MAD, Wavelet) are methods that work only under architecture- and attack-agnostic setting. For comparison, we add (in gray shading) two variants of Mahalanobis method [16]: the regular and a class agnostic variant (marked with *). **Bold** - best, underline - second best.

Model	Method	FGSM	PGD	Auto Attack	BIM	CW	Deep Fool	Square	Avg.
Wide ResNet	A^4D (Ours)	99.64	<u>99.67</u>	99.75	<u>99.44</u>	78.03	77.55	93.61	92.53
	MAD	<u>85.80</u>	79.08	82.36	62.94	50.02	49.98	<u>81.45</u>	70.23
	Wavelet	99.64	97.43	97.68	82.88	50.48	50.11	70.28	78.36
	Mahalanobis	67.10	99.91	<u>99.24</u>	99.76	<u>55.26</u>	<u>55.44</u>	62.08	76.97
	Mahalanobis*	67.10	82.91	81.60	78.64	54.65	55.10	50.02	67.15
Res- Net34	A^4D (Ours)	<u>94.70</u>	98.01	<u>96.79</u>	96.27	86.27	85.64	95.57	93.32
	MAD	85.55	78.85	82.20	63.34	50.17	50.00	<u>82.94</u>	70.43
	Wavelet	99.58	<u>97.42</u>	97.63	<u>82.49</u>	50.46	50.02	70.46	78.29
	Mahalanobis	91.94	91.52	85.05	77.27	<u>70.99</u>	<u>68.58</u>	67.22	<u>78.94</u>
	Mahalanobis*	91.94	13.14	29.33	18.59	38.57	38.24	42.61	38.92
Deit- Small	A^4D (Ours)	99.24	98.16	98.39	97.76	77.31	78.62	97.84	92.47
	MAD	73.46	71.07	85.95	83.46	50.68	50.46	63.52	68.37
	Wavelet	<u>95.99</u>	<u>95.05</u>	<u>93.69</u>	<u>90.36</u>	50.92	50.62	56.88	<u>76.22</u>
	Mahalanobis	61.45	61.94	-	62.53	<u>63.82</u>	<u>63.55</u>	<u>69.08</u>	63.72
	Mahalanobis*	61.45	60.59	-	60.51	47.06	48.27	51.92	54.96
Conv- next tiny	A^4D (Ours)	93.80	95.38	97.87	83.18	<u>62.83</u>	<u>63.47</u>	98.24	84.97
	MAD	77.73	69.64	72.29	57.12	50.10	50.01	76.65	64.79
	Wavelet	<u>94.07</u>	87.33	<u>87.68</u>	69.88	50.39	50.24	70.44	72.86
	Mahalanobis	96.46	<u>97.28</u>	-	<u>97.65</u>	74.64	75.13	<u>91.98</u>	88.85
	Mahalanobis*	96.46	98.29	-	98.27	59.83	57.48	90.25	83.43

5.4 Limitations

Despite its strengths, our method has several limitations. First, its performance depends on the choice of textual prompts and the representation quality of CLIP. In our method, the prompts are chosen manually based on their performance on the validation set. Not all semantically plausible noise-related prompts are suitable for this task, as shown in Figure 4, where the straightforward general sentence “A noisy image” is highly non-discriminative for all attacks. In order to define a dictionary with powerful discriminative capabilities, we aim to select sentences that are both discriminative and low-correlated, as illustrated in Fig. 5. However, a more systematic procedure for prompt selection could be developed. There exists a natural trade-off. On the one hand, learning the prompts may lead to overfitting, which contradicts the goal of maintaining a general zero-shot setting. On the other hand, manually selected prompts may not generalize equally well across different attack types. As a compromise, we select semantically relevant prompts with low correlation among them to encourage diversity. Finally, our approach assumes access to clean training data that is representative of the test distribution. Improving robustness under distribution shift and

Table 4: Ablation on aggregation functions. Zero-shot adversarial detection performance (AU-ROC) on validation set From tiny-ImageNet and DeiT-Small classifier. **Bold** - best, underline - second best.

Method	FGSM	PGD	AutoAttack	BIM	CW	DeepFool	Square	Avg.
Max Percentile	99.25	98.54	98.63	97.99	74.70	76.22	<u>95.35</u>	<u>91.53</u>
Min Percentile	93.07	89.84	89.73	85.66	69.01	69.75	88.05	83.58
Mean Percentile	98.63	97.44	97.61	96.81	<u>75.19</u>	<u>76.86</u>	94.77	91.04
PCA Aggregation	<u>99.24</u>	<u>98.16</u>	<u>98.39</u>	<u>97.76</u>	77.31	78.62	97.84	92.47

developing principled prompt selection strategies remain promising directions for future work.

6 Conclusions

We proposed a zero-shot adversarial attack detection framework based on prompt-image similarity representations derived from a vision-language model. The method learns a compact representation of clean data using only clean samples and enables test-time detection without prior knowledge of the attack type or the underlying classifier.

By leveraging the expressive representations of vision-language models, the proposed approach can identify subtle perturbations that induce shifts in the embedding space while remaining visually inconspicuous. At the same time, detection requires only a single forward pass per image followed by lightweight similarity-based computations, resulting in an efficient and practical solution. Overall, this work highlights the potential of vision-language models as a general and efficient tool for adversarial detection.

Acknowledgments

We would like to acknowledge support by the Israel Science Foundation (Grant 1472/23) and by the Ministry of Innovation, Science and Technology (Grant 8801/25).

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European conference on computer vision. pp. 484–501. Springer (2020)
2. Betser, R., Levi, M.Y., Gilboa, G.: Whitened clip as a likelihood surrogate of images and captions. In: 42nd International conference on machine learning (2025)
3. Cao, Y., Zhang, J., Frittoli, L., Cheng, Y., Shen, W., Boracchi, G.: Adapclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In: European Conference on Computer Vision. pp. 55–72. Springer (2024)

4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)
5. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: International Conference on Machine Learning. pp. 1310–1320. PMLR (2019)
6. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International Conference on Machine Learning. pp. 2206–2216. PMLR (2020)
7. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: Vqgan-clip: Open domain image generation and editing with natural language guidance. In: European Conference on Computer Vision. pp. 88–105. Springer (2022)
8. Danesh, W., Sapireddy, S.R., Rahman, M.: Understanding and detecting adversarial examples in IoT networks: A white-box analysis with autoencoders. *Electronics* **14**(15), 3015 (2025)
9. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410 (2017)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
12. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15262–15271 (2021)
13. Kapp, A., Hoffmann, E., Weigmann, E., Mihaljević, H.: StreetSurfaceVis: a dataset of crowdsourced street-level imagery annotated by road surface type and quality. *Scientific Data* **12**(1), 92 (2025). <https://doi.org/10.1038/s41597-024-04295-9>, <https://doi.org/10.1038/s41597-024-04295-9>
14. Kim, H.: Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950 (2020)
15. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial intelligence safety and security, pp. 99–112. Chapman and Hall/CRC (2018)
16. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems* **31** (2018)
17. Levi, M.Y., Gilboa, G.: The double ellipsoid geometry of clip. In: Proceedings of the 42nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 267. PMLR, Vancouver, Canada (2025)
18. Li, Q., Wu, C., Chen, J., Zhang, Z., He, K., Du, R., Wang, X., Zhao, Q., Liu, Y.: Privacy-preserving universal adversarial defense for black-box models. *IEEE Transactions on Information Forensics and Security* (2025)
19. Liang, H., He, E., Zhao, Y., Jia, Z., Li, H.: Adversarial attack and defense: A survey. *Electronics* **11**(8), 1283 (2022)
20. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986 (2022)
21. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neuro-computing* **508**, 293–304 (2022)

22. Ma, W., Zhang, X., Yao, Q., Tang, F., Wu, C., Li, Y., Yan, R., Jiang, Z., Zhou, S.K.: Aa-clip: Enhancing zero-shot anomaly detection via anomaly-aware clip. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 4744–4754 (2025)
23. Ma, X., Li, B., Wang, Y., Erfani, S.M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M.E., Bailey, J.: Characterizing adversarial subspaces using local intrinsic dimensionality. arXiv preprint arXiv:1801.02613 (2018)
24. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
25. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. pp. 135–147 (2017)
26. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267 (2017)
27. mnmostafa, Ali, M.: Tiny imagenet. <https://kaggle.com/competitions/tiny-imagenet> (2017), kaggle
28. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
29. Nazeri, A., Hafez, W.: Entropy-based non-invasive reliability monitoring of convolutional neural networks. arXiv preprint arXiv:2508.21715 (2025)
30. Papernot, N., McDaniel, P.: Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv preprint arXiv:1803.04765 (2018)
31. Peng, Z., Xu, Z., Zeng, Z., Wen, C., Huang, Y., Yang, M., Tang, F., Shen, W.: Understanding fine-tuning clip for open-vocabulary semantic segmentation in hyperbolic space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4562–4572 (2025)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
33. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International conference on machine learning. pp. 8821–8831. Pmlr (2021)
34. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* **35**, 36479–36494 (2022)
35. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! *Advances in neural information processing systems* **32** (2019)
36. Stenhuis, R., Liu, D., Qiao, Y., Conti, M., Panaousis, M., Liang, K.: Meetsafe: enhancing robustness against white-box adversarial examples. *Frontiers in Computer Science* **7**, 1631561 (2025)
37. Sultan, M., Jacobs, L., Stylianou, A., Pless, R.: Exploring clip for real world, text-based image retrieval. In: 2023 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). pp. 1–6. IEEE (2023)
38. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)

39. Wei, Y., Cao, Y., Zhang, Z., Peng, H., Yao, Z., Xie, Z., Hu, H., Guo, B.: iclip: Bridging image classification and contrastive language-image pre-training for visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2776–2786 (2023)
40. Weng, Z., Yang, X., Li, A., Wu, Z., Jiang, Y.G.: Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In: International conference on machine learning. pp. 36978–36989. PMLR (2023)
41. Wu, S., Zhang, W., Xu, L., Jin, S., Li, X., Liu, W., Loy, C.C.: Clipself: Vision transformer distills itself for open-vocabulary dense prediction. arXiv preprint arXiv:2310.01403 (2023)
42. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
43. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
44. Zhang, Y., Liang, P.: Defending against whitebox adversarial attacks via randomized discretization. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 684–693. PMLR (2019)
45. Zhou, Q., Pang, G., Tian, Y., He, S., Chen, J.: Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. arXiv preprint arXiv:2310.18961 (2023)