Don't Lag, RAG: Training-Free Adversarial Detection Using RAG

Roie Kazoom*
Electrical and Computers Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
roieka@post.bgu.ac.il

Raz Lapid
DeepKeep.ai
Tel Aviv, Israel
raz.lapid@deepkeep.ai

Moshe Sipper
Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel
sipper@post.bgu.ac.il

Ofer Hadar
Electrical and Computers Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
hadar@post.bgu.ac.il

Abstract

Adversarial patch attacks pose a major threat to vision systems by embedding localized perturbations that mislead deep models. Traditional defense methods often require retraining or fine-tuning, making them impractical for real-world deployment. We propose a training-free Visual Retrieval-Augmented Generation (VRAG) framework that integrates Vision-Language Models (VLMs) for adversarial patch detection. By retrieving visually similar patches and images that resemble stored attacks in a continuously expanding database, VRAG performs generative reasoning to identify diverse attack types-all without additional training or fine-tuning. We extensively evaluate open-source large-scale VLMs-including Qwen-VL-Plus, Qwen2.5-VL-72B, and UI-TARS-72B-DPO-alongside Gemini-2.0, a closed-source model. Notably, the open-source UI-TARS-72B-DPO model achieves up to 95% classification accuracy, setting a new state-of-the-art for open-source adversarial patch detection. Gemini-2.0 attains the highest overall accuracy, 98%, but remains closed-source. Experimental results demonstrate VRAG's effectiveness in identifying a variety of adversarial patches with minimal human annotation, paving the way for robust, practical defenses against evolving adversarial patch attacks.

1 Introduction

Deep learning models, particularly convolutional neural networks (CNNs) [23, 18, 47] and vision transformers (ViTs) [12], have demonstrated remarkable success in computer vision tasks such as object detection [15, 43, 42], image classification [23, 12], and segmentation [34, 46]. However, despite advances, these models remain highly vulnerable to adversarial attacks [49, 24, 25, 2, 16, 35, 50, 27], where small perturbations or carefully crafted patches manipulate predictions.

Adversarial patch attacks [6, 28, 33, 54] introduce localized perturbations that persist across different transformations, making them significantly more challenging to mitigate using conventional defense mechanisms [53]. Unlike traditional adversarial perturbations that introduce subtle noise across an image, adversarial patches are structured, high-magnitude perturbations, which are often physically

*Corresponding author: roieka@post.bgu.ac.il

39th Conference on Neural Information Processing Systems VLM4RWD (NeurIPS 2025).

Figure 1: Illustration of three different settings for detecting adversarial patches. (Left) The zero-shot baseline, in which the model is directly prompted to determine if the image is adversarial but incorrectly concludes it is benign. (Center) Our VRAG-based approach on a benign image; as the database does not contain benign exemplars, no relevant references are retrieved. Consequently, the classification relies solely on the prompt content and remains accurate. (Right) Our VRAG-based approach on an adversarial image, which leverages relevant references from the database to enhance the prompt, ultimately yielding a correct detection of the adversarial patch.

realizable. These patches can be printed, placed in real-world environments, and still cause misclassification or mislocalization in deployed deep learning models. Their adversarial effect remains robust under different lighting conditions, transformations, and occlusions, allowing them to be successfully deployed in real world scenarios [32, 11]. Furthermore, retraining-based defenses require extensive, labeled adversarial data, which is expensive to obtain and generalizes poorly to novel attack strategies [53].

Traditional adversarial detection methods typically fall into one of three categories, (1) supervised learning-based defenses, (2) unsupervised defenses and (3) adversarial training. Supervised learning-based defenses [39, 38] use deep learning classifiers trained on labeled adversarial and non-adversarial samples. These methods are data-dependent and do not adapt well to adversarial attacks outside the training distribution. Unsupervised defenses [58, 37, 48, 36], typically rely on analyzing the intrinsic structure or distribution of unlabeled data to detect anomalous inputs. For example, Feature Squeezing [58] reduces input dimensionality (e.g., through bit-depth reduction or smoothing) to reveal suspicious high-frequency artifacts; [37] use deep generative models to flag inputs with high reconstruction error as potential adversarial samples. Although these methods can detect novel or previously unseen attack strategies without relying on adversarial labels, they often require carefully chosen hyperparameters and remain vulnerable to adaptive attacks that mimic the statistics of benign inputs. In contrast to the supervised detection methods, which *separately classify* inputs as adversarial or benign, *adversarial training* [35, 26] augments the training data with adversarial examples to directly improve model robustness. Rather than solely learning to detect adversarial inputs, this approach modifies the model parameters and decision boundaries to make correct classification more likely under attack. However, adversarial training is computationally expensive and risks overfitting to specific attack types, leading to weaker defenses against unseen attacks [29].

In this paper, we introduce a retrieval-augmented adversarial patch detection framework that dynamically adapts to evolving threats without necessitating retraining. The method integrates visual retrieval-augmented generation (VRAG) with a vision-language model (VLM) for context-aware detection. As illustrated in Figure 1, visually similar patches are retrieved from a precomputed database using semantic embeddings from grid-based image regions, and structured natural language prompts guide the VLM to classify suspicious patches.

This paper makes the following contributions:

1. A training-free retrieval-based pipeline that dynamically matches adversarial patches against a precomputed (and expandable) database.
2. The integration of existing VLMs with generative reasoning for context-aware patch detection through structured prompts.
3. A comprehensive evaluation demonstrating robust detection across diverse adversarial patch scenarios, all without additional training or fine-tuning.

Experimental results confirm that our retrieval-augmented detection approach not only outperforms traditional classifiers, but also achieves state-of-the-art detection across a variety of threat scenarios. This method offers higher accuracy and reduces dependence on labeled adversarial datasets, underscoring the practicality of incorporating retrieval-based strategies alongside generative reasoning to develop scalable, adaptable defenses for real-world security applications [45].

2 Related Work

Adversarial attacks exploit neural network vulnerabilities through carefully crafted perturbations. Early works focused on small, imperceptible ℓ_p-bounded perturbations such as FGSM [16] and PGD [35]. In contrast, *adversarial patch attacks* apply localized, high-magnitude changes that remain effective under transformations and pose a threat in real-world scenarios [19, 32, 11].

Defenses fall into *reactive* and *proactive* categories. Reactive methods like JPEG compression [13] and spatial smoothing [57] attempt to remove adversarial patterns at inference time but struggle against adaptive attacks. Diffusion-based methods, such as DIFFender [20] and purification models [30], leverage generative models to restore clean content but are often computationally intensive.

Another line of work focuses on *patch localization and segmentation*, e.g., SAC [31], which detects and removes patches using segmentation networks. These approaches are limited by their reliance on training and struggle with irregular or camouflaged patches. PatchCleanser [55] offers certifiable robustness but assumes geometrically simple patches. Proactive defenses like adversarial training [53] aim to increase robustness through exposure to adversarial examples. While effective against known attacks, they generalize poorly and are resource-intensive.

We propose a retrieval-augmented framework that detects a wide range of patch types-including irregular and naturalistic ones-without degrading input quality or relying on segmentation or geometric assumptions. Our method leverages a diverse patch database and vision-language reasoning to dynamically adapt to unseen attacks.

3 Preliminaries

We briefly review core paradigms relevant to our defense framework: vision-language foundation models, zero- and few-shot learning, adversarial attacks and defenses, and RAG.

3.1 Vision-Language Foundation Models and Zero- and Few-Shot Learning

Foundation models leverage large-scale transformer architectures and self-attention to learn general-purpose representations from massive image-text data. A typical *VLM* consists of two encoders, f_θ for images I and g_ϕ for text T, projecting them into a shared embedding space:

$$E_I = f_\theta(I), \quad E_T = g_\phi(T), \quad S(I,T) = \frac{E_I \cdot E_T}{\|E_I\|\|E_T\|}. \tag{1}$$

Models like CLIP [41] and Flamingo [1] align image-text pairs via contrastive objectives, enabling flexible *zero-shot* capabilities:

$$g(I, Q) \to A, \tag{2}$$

where Q is a textual query and A is the inferred label without explicit task-specific training. *Few-shot learning* refines zero-shot by supplying a small support set $\{(I_1, y_1), \ldots, (I_k, y_k)\}$:

$$g\big(I, Q \mid \{(I_i, y_i)\}_{i=1}^{k}\big) \to A, \tag{3}$$

allowing adaptation to novel tasks with limited labeled data.

3.2 Adversarial Attacks and Defense Strategies

Adversarial Attacks. Formally, an adversary seeks a perturbation δ subject to $\|\delta\|_p \leq \epsilon$ that maximizes a loss function ℓ for a model f_θ with true label y:

$$\delta^* = \arg\max_{\|\delta\|_p \leq \epsilon} \ell\big(f_\theta(I + \delta), y\big). \tag{4}$$

Patch-based attacks instead replace a localized region using a binary mask $M \in \{0, 1\}^{H \times W}$:

$$I' = I \odot (1 - M) + P \odot M, \tag{5}$$

where P is a high-magnitude patch. Such localized perturbations remain visually inconspicuous in many practical settings [19, 32, 11].

Preprocessing and Detection. A common defense strategy is to apply a transformation $g(\cdot)$ to I', yielding $g(I')$, with the goal of suppressing adversarial noise (e.g., blurring, smoothing [22]). Detection can be formulated by a function $D\big(g(I')\big) \in \{0, 1\}$ that flags anomalous inputs based on statistical or uncertainty-based criteria [7].

Generative Reconstruction. Diffusion-based defenses [20] iteratively denoise adversarial inputs by reversing a noisy forward process:

$$x_t = \sqrt{\alpha_t}\, x_{t-1} \; + \; \sqrt{1 - \alpha_t}\, \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I), \tag{6}$$

often guided by patch localization [31]. Although effective, these approaches can falter against unseen attacks or large patch perturbations, making robust generalization challenging in practice.

3.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) integrates external knowledge into a generative model to improve both its generative capacity and semantic coherence. Formally, given a query Q, the model retrieves the top-k most relevant documents or embeddings R_k from a database $\mathcal{D}$:

$$R_k = \arg\max_{R_i \in \mathcal{D}} S(Q, R_i), \tag{7}$$

where $S(\cdot, \cdot)$ is a similarity function. The query Q is then combined with R_k within a generative function:

$$A = G(Q, R_k). \tag{8}$$

In our approach, this retrieval phase facilitates access to known adversarial patches, thereby enabling a more robust generative reasoning process. By incorporating historical data on diverse attack patterns, RAG-based defenses can dynamically adapt to novel threats while sustaining high efficacy against existing adversaries.

4 Methodology

This section details our VRAG-based approach for adversarial patch detection using a vision-language model. We first describe the construction of a comprehensive adversarial patch database (§4.1), then present our end-to-end detection pipeline (§4.2), and finally discuss how the framework generalizes to diverse patch shapes (§A.6). To enable scalability, we parallelize patch embedding and augmentation—see Appendix A.1 for runtime benchmarks across varying numbers of workers.

4.1 Database Creation

To handle a wide variety of adversarial patch attacks, we build a large-scale database of patched images and their corresponding patch embeddings. We aggregate patches generated by SAC [31], BBNP [28], and standard adversarial patch attacks [6], placing each patch onto diverse natural images at random positions and scales. This process, summarized in 1, ensures that the database spans different patch configurations and visual contexts.

Algorithm 1 Adversarial Patch Database Creation with Positional Augmentation

Input: Set of patches $\{P_i\}_{i=1}^{m}$, set of natural images $\{I_j\}_{j=1}^{q}$, embedding model f, grid size $n \times n$, number of placement variations A
Output: Database $\mathcal{D}$
Initialize database $\mathcal{D} \leftarrow \emptyset$
for $i = 1$ to m **do**
    Compute patch embedding $E_{P_i} = f(P_i)$
    Store (P_i, E_{P_i}) in $\mathcal{D}$
    for $j = 1$ to q **do**
        for $a = 1$ to A **do**
            Randomly select position (x_a, y_a) in image I_j
            Apply patch P_i at (x_a, y_a) to obtain patched image $I_j^{(a)}$
            Divide $I_j^{(a)}$ into grid cells $\{C_{j,k}^{(a)}\}_{k=1}^{n^2}$
            for $k = 1$ to n^2 **do**
                Compute embedding $E_{j,k}^{(a)} = f(C_{j,k}^{(a)})$
                if $C_{j,k}^{(a)}$ overlaps with P_i **then**
                    Store $(C_{j,k}^{(a)}, E_{j,k}^{(a)})$ in $\mathcal{D}$
                end if
            end for
        end for
    end for
end for
return $\mathcal{D}$

Concretely, each patched image is subdivided into an $n \times n$ grid, yielding localized regions $\{C_1, \ldots, C_{n^2}\}$ that spatially partition the image. For each region C_i, we compute a dense visual embedding using a pre-trained vision encoder $f(\cdot)$:

$$E_{C_i} = f(C_i),$$

which captures high-level semantic and structural features of the corresponding image patch. In parallel, we encode each adversarial patch P_j into its own latent representation $E_{P_j} = f(P_j)$ to ensure embeddings are in the same feature space. These patch embeddings act as *keys*, while the embeddings of overlapping regions serve as their corresponding *values* in a key-value database. This design enables efficient and scalable nearest-neighbor retrieval at inference time, allowing the system to match visual evidence in test images with known adversarial patterns from the database.

4.2 VRAG-Based Detection Pipeline

System Overview. Our detection system (illustrated in Figure 2) identifies adversarial patches in a query image by leveraging the patch database as retrieval context for a vision-language model. The process involves four main steps:

1. **Image Preprocessing:** Divide the input image I into an $n \times n$ grid of regions $\{C_1, \ldots, C_{n^2}\}$ to enable localized inspection of each part of the image.
2. **Feature Extraction:** Encode each region C_i into an embedding $E_i = f(C_i)$ using a pre-trained vision encoder (e.g., CLIP). These embeddings capture high-level semantic features.

3. **Retrieval Step:** For each E_i, perform a nearest-neighbor search in the patch database $\mathcal{D}$. Retrieve the top-k most similar patch embeddings to form a context set $\mathcal{R}_i = \text{Top-}k(\{d(E_i, E_{P_j})\})$. Appendix A.3 presents an ablation study comparing cosine similarity with alternative distance metrics for this retrieval step.
4. **Generative Reasoning with a VLM:** Combine each region C_i with its retrieved examples $\mathcal{R}_i$ and short textual cues to construct a multimodal prompt. This prompt is passed to a vision-language model $g(\cdot)$ to answer:

$$g(\mathcal{C}_i) \rightarrow \text{“Does this region contain an adversarial patch?”}$$

We summarize the overall detection procedure in 2.

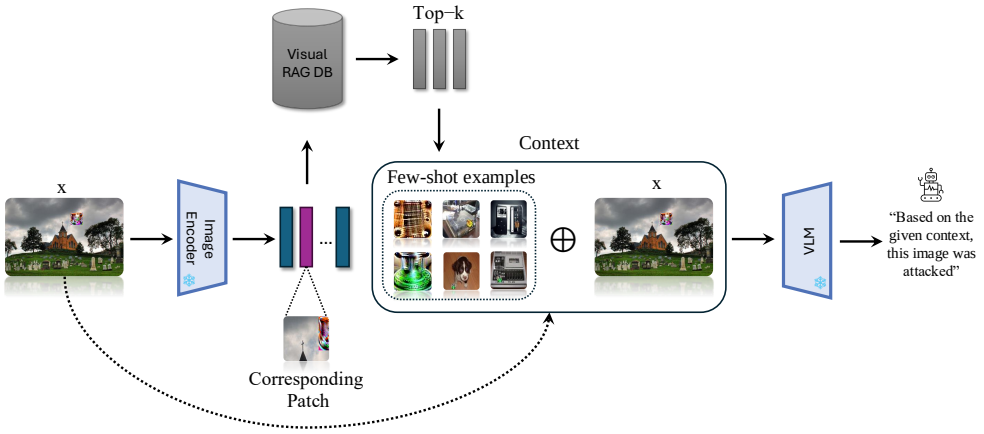


Figure 2: Overview of our VRAG framework for adversarial patch detection. Given a query image, we extract grid-based embeddings and retrieve the top-k visually similar adversarial patches from our database. These patches and their associated attacked images form a few-shot context for a vision-language model that decides whether the query contains an adversarial patch.

Algorithm 2 Adversarial Patch Detection via VRAG

Input: Image I, VLM $\mathcal{V}$, Database $\mathcal{D}$, Embedding function f, threshold τ, top-m patches, top-k images
Output: Decision: `Attacked` or `Not Attacked`
Divide I into grid cells $\{C_i\}_{i=1}^{n^2}$, compute embeddings $E_i = f(C_i)$
for each E_i **do**
    Compute max similarity $S_i = \max_{E_d \in \mathcal{D}} \frac{E_i \cdot E_d}{\|E_i\| \|E_d\|}$
end for
Select candidates $\mathcal{C} = \{C_i \mid S_i \geq \tau\}$, choose top-$m$ patches
Retrieve top-k similar attacked images from $\mathcal{D}$
Build context $\mathcal{T}$ with top-m patches and top-k images as examples
Query VLM with $\mathcal{T}$: $R = \mathcal{V}(\mathcal{T}, I)$
return $R \in \{\texttt{Attacked}, \texttt{Not Attacked}\}$

Decision Mechanism (Zero-Shot and Few-Shot). After retrieving similar patches and attacked images, the VLM is prompted to judge the query image under zero-shot or few-shot conditions:

- *Zero-Shot Detection:* The model relies on pre-trained knowledge and textual prompts to classify each region C_i as adversarial or benign, without additional fine-tuning.
- *Few-Shot Adaptation:* A small, labeled set of adversarial examples, denoted as $\{A_i\}$, along with their corresponding patches $\{P_i\}$, is incorporated into the retrieved context to refine the model's decision-making process. This integration enhances the model's robustness to previously unseen attacks by explicitly exposing the VLM to representative instances of patch-induced behavior.

A sample query prompt for the VLM might be:

```
"Here are examples of adversarial patches: [Patch 1], [Patch 2].
Here are images that contain these patches: [Image 1], [Image
2]. Based on this context, does the following image contain an
adversarial patch? Answer 'yes' or 'no'."
```

The model's answer is then used to decide whether the image is `Attacked` or `Not Attacked`.

Optimal Threshold Selection. We determine the optimal threshold based on ROC-AUC analysis of cosine similarity scores computed from embedding vectors. Specifically, the optimal cosine similarity threshold identified was 0.77, providing the best trade-off between sensitivity and specificity. We observed that for thresholds approaching 1.0, the similarity criterion becomes overly permissive, resulting in nearly every image retrieving similar images, thereby substantially increasing the false-positive rate.

5 Experimental Evaluation

We conduct extensive experiments to assess the robustness and efficiency of our adversarial patch detection framework across diverse datasets, models, attack types, and defenses, simulating realistic deployment scenarios.

Vision Language Models. For generative reasoning, we use several VLMs $g(\cdot)$, including *Qwen-VL-Plus* [8], *Qwen2.5-VL-Instruct* [9], *UI-TARS-72B-DPO* [44], and *Gemini* [10]. These were chosen for their strong multimodal reasoning in zero- and few-shot settings. While *Gemini 2.0* yields the highest accuracy, it is proprietary. *UI-TARS-72B-DPO*, meanwhile, offers competitive performance and sets a strong benchmark among open-source models.

Classification Models. To evaluate the impact of adversarial patches across diverse architectures, we consider four representative image classification models: (1) *ResNet-50*[18], (2) *ResNeXt-50*[56], (3) *EfficientNet-B0*[51], and (4) *ViT-B/16*[12]. These models span both convolutional and transformer-based paradigms and offer a clear comparison across varying robustness profiles and architectural biases. For all models, we report clean and attacked accuracies under each defense method, using the same attack configuration and patch size distribution.

Datasets and Attacks. We evaluate on both synthetic and real-world patch benchmarks: (1) *ImageNet-Patch* [40], a 50/50 balanced dataset of attacked and clean ImageNet samples, where attacks are applied to exactly 50% of the data to ensure balanced evaluation; and (2) *APRICOT* [31], a real-world dataset of 873 images, each containing a physically applied adversarial patch.

We test two strong attacks: the classical adversarial patch [6] targeting CNNs, and PatchFool [14] targeting vision transformers. Patches are randomly placed and vary in size from 25×25 to 65×65.

Defense Mechanisms. We compare against several baselines: (1) JPEG compression [13], (2) Spatial smoothing [57], (3) SAC [31], and (4) DIFFender [20], a recent diffusion-based approach. We also evaluate a retrieval-only baseline that flags regions as adversarial based on visual similarity, without using VLM reasoning.

Evaluation Protocol. On *ImageNet-Patch*, we report classification accuracy over a balanced 50/50 clean/attacked split. On *APRICOT*, we report binary accuracy (presence vs. absence of a patch) across three settings: (1) *Clean*, (2) *Undefended*, and (3) *Defended*. Candidate regions are retrieved using top-$k = 2$ cosine similarity and verified via VLM prompts. Thresholds are calibrated on a held-out validation set to ensure fair comparisons across all methods.

6 Results

Table 1 presents the accuracy of various defense mechanisms on the APRICOT dataset [5] under adversarial patch attacks of sizes from 25×25 to 65×65. Traditional defenses such as JPEG compression [13], spatial smoothing [57], and SAC [31] show only modest improvements, while DIFFender [20] achieves stronger robustness and better accuracy across all patch sizes. Our method consistently outperforms 0-shot baselines and scales better with higher-strength attacks. Even the 0-shot version attains competitive accuracy, while the 4-shot configuration yields substantial gains,

maintaining high accuracy under challenging conditions, as shown in Appendix 7. Table 2 reports defense accuracy against adversarial patch attacks of varying sizes. Performance drops sharply without defense. Traditional methods-JPEG compression [13], spatial smoothing [58], and SAC [31]-show limited robustness, while DIFFender [20] performs better via generative reconstruction. Our retrieval-only baseline surpasses these, and the full VLM-based method achieves the best results, with the 4-shot variant nearly restoring clean accuracy, demonstrating the effectiveness of retrieval-augmented generative reasoning. Extensive evaluations on ImageNet-Patch and APRICOT demonstrate strong robustness across patch sizes and attack methods, outperforming JPEG compression [13], spatial smoothing [57], SAC [31], and DIFFender [20]. Our system achieves up to 98% detection accuracy while maintaining performance under severe threats. Appendix 6 and A.6 present qualitative and generalization analyses across diverse patch shapes.

Table 1: Accuracy (%) on APRICOT [5] with adversarial patches of varying sizes. Methods are evaluated in 0-shot (0S), 2-shot (2S), and 4-shot (4S) settings; methods without few-shot use show "–". Gray indicates the best 0S result, underline the second-best overall, and **bold** the best overall.

Method	25×25			50×50			55×55			65×65		
	0S	2S	4S	0S	2S	4S	0S	2S	4S	0S	2S	4S
Undefended	34.59	–	–	32.18	–	–	30.24	–	–	28.55	–	–
JPEG [13]	29.35	–	–	32.53	–	–	35.28	–	–	41.11	–	–
Spatial Smoothing [57]	33.56	–	–	36.19	–	–	39.17	–	–	42.26	–	–
SAC [31]	45.93	–	–	48.22	–	–	49.14	–	–	52.80	–	–
DIFFender [20]	65.06	–	–	66.32	–	–	68.61	–	–	70.90	–	–
Baseline	56.81	–	–	59.56	–	–	60.59	–	–	69.64	–	–
Ours (Qwen-VL-Plus)	45.37	76.18	87.64	46.40	77.90	88.78	47.55	79.62	90.50	50.98	81.91	92.22
Ours (Qwen2.5-VL-72B)	47.37	78.18	89.64	48.40	79.90	90.78	49.55	81.62	92.50	52.98	83.91	94.22
Ours (UI-TARS-72B-DPO)	49.37	80.18	91.64	50.40	81.90	92.78	51.55	83.62	94.50	54.98	85.91	96.22
Ours (Gemini)	56.24	82.59	**93.92**	57.16	85.11	**96.33**	58.76	86.94	**96.79**	63.12	90.26	**97.93**

Table 2: Accuracy (%) of four models under adversarial patch attacks of varying sizes. Each method is evaluated under three configurations: 0-shot (0S), 2-shot (2S), and 4-shot (4S), reflecting increasing levels of visual context provided to the vision-language model. For methods that do not support few-shot adaptation, results for 2S and 4S are omitted and marked with "–". Gray indicates the best-performing method in the 0-shot setting, underline highlights the second-best overall result across all configurations, and **bold** denotes the highest overall accuracy. This presentation enables a clear comparison of zero- and few-shot performance across varying patch sizes and models.

Model	**Method**	**Clean**	25×25			50×50			55×55			65×65		
			0S	2S	4S	0S	2S	4S	0S	2S	4S	0S	2S	4S
ResNet-50 [18]	Undefended	97.50	7.50	–	–	9.25	–	–	8.75	–	–	6.95	–	–
	JPEG [13]		50.75	–	–	51.75	–	–	49.25	–	–	49.00	–	–
	Spatial Smoothing [57]		55.50	–	–	58.25	–	–	55.25	–	–	50.75	–	–
	SAC [31]		64.75	–	–	66.75	–	–	68.00	–	–	69.50	–	–
	Baseline		58.50	–	–	59.75	–	–	62.00	–	–	62.50	–	–
	Ours (Qwen-VL-Plus)		49.75	70.00	85.25	54.00	73.00	86.50	62.50	75.00	87.25	79.00	79.50	88.00
	Ours (Qwen2.5-VL-72B)		55.25	82.00	88.25	60.00	84.00	89.25	79.75	86.00	90.50	91.00	91.25	91.50
	Ours (UI-TARS-72B-DPO)		54.50	83.00	89.50	55.50	87.75	90.50	57.50	86.25	89.75	57.50	87.50	94.00
	Ours (Gemini)		56.25	87.25	**93.25**	58.50	89.75	**93.75**	59.75	90.25	**96.25**	60.25	91.25	**99.25**
ResNeXt-50 [56]	Undefended	97.50	9.25	–	–	11.00	–	–	10.75	–	–	8.95	–	–
	JPEG [13]		48.75	–	–	50.75	–	–	47.75	–	–	46.50	–	–
	Spatial Smoothing [57]		55.75	–	–	57.50	–	–	55.75	–	–	50.25	–	–
	SAC [31]		64.75	–	–	66.25	–	–	68.00	–	–	66.75	–	–
	Baseline		56.50	–	–	58.50	–	–	60.25	–	–	61.75	–	–
	Ours (Qwen-VL-Plus)		48.25	68.50	83.00	52.00	71.25	84.50	58.00	72.75	85.25	74.25	77.00	86.25
	Ours (Qwen2.5-VL-72B)		53.25	78.25	85.75	58.50	80.75	87.00	76.00	84.00	88.25	89.25	90.00	90.75
	Ours (UI-TARS-72B-DPO)		52.50	80.75	85.75	55.25	85.00	89.25	55.25	86.25	91.00	59.25	84.75	93.25
	Ours (Gemini)		55.50	85.00	**91.25**	57.75	87.50	**92.75**	58.75	88.50	**94.75**	60.75	89.75	**98.50**
EfficientNet [51]	Undefended	95.50	24.25	–	–	25.75	–	–	24.00	–	–	21.50	–	–
	JPEG [13]		51.00	–	–	53.75	–	–	50.75	–	–	49.25	–	–
	Spatial Smoothing [57]		60.50	–	–	63.25	–	–	61.75	–	–	57.50	–	–
	SAC [31]		58.25	–	–	60.75	–	–	63.25	–	–	67.25	–	–
	Baseline		54.75	–	–	56.75	–	–	58.25	–	–	61.00	–	–
	Ours (Qwen-VL-Plus)		50.25	69.25	84.00	53.00	72.25	85.50	59.50	74.00	86.25	76.00	78.75	87.75
	Ours (Qwen2.5-VL-72B)		54.50	79.50	87.00	59.25	82.00	89.00	78.25	85.00	90.50	90.50	91.00	92.00
	Ours (UI-TARS-72B-DPO)		49.75	80.50	85.25	52.25	83.00	88.75	54.75	82.75	91.00	57.75	86.00	95.00
	Ours (Gemini)		53.00	84.25	**91.25**	55.50	85.75	**93.50**	57.00	88.00	**95.75**	59.75	89.75	**97.50**
ViT-B-16 [14]	Undefended	97.75	27.75	–	–	29.25	–	–	27.00	–	–	24.25	–	–
	JPEG [13]		57.75	–	–	58.75	–	–	55.50	–	–	51.00	–	–
	Spatial Smoothing [57]		66.75	–	–	67.25	–	–	64.00	–	–	61.25	–	–
	SAC [31]		63.25	–	–	64.75	–	–	65.75	–	–	69.25	–	–
	Baseline		59.50	–	–	61.50	–	–	62.75	–	–	64.00	–	–
	Ours (Qwen-VL-Plus)		51.25	69.50	84.25	55.00	72.50	85.75	60.50	76.00	86.75	74.00	79.00	87.25
	Ours (Qwen2.5-VL-72B)		56.75	78.75	87.00	60.75	81.00	88.75	78.00	84.50	90.50	90.25	91.00	91.75
	Ours (UI-TARS-72B-DPO)		53.25	82.00	89.75	54.75	84.25	91.00	56.75	85.50	93.25	59.50	88.75	95.25
	Ours (Gemini)		58.75	86.75	**93.50**	60.75	89.00	**95.25**	61.25	90.75	**98.75**	63.00	93.00	**99.00**

Limitations and Future Work. Our method assumes access to a representative patch database. Future work will focus on automatic patch discovery, uncertainty quantification in VLM outputs, and improving inference speed for large-scale deployment.

Conclusion. The proposed VRAG-based framework combines retrieval-augmented search with generative reasoning to deliver robust, scalable, and training-free adversarial patch detection. It achieves high accuracy, generalizes across patch types, and effectively detects unseen attacks.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736, 2022.

[2] Tal Alter, Raz Lapid, and Moshe Sipper. On the robustness of kolmogorov-arnold networks: An adversarial perspective. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=uafxqhImPM`.

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond, 2024. URL `https://openreview.net/forum?id=qrGjFJVl3m`.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] A. Braunegg, A. Chakraborty, M. Krumdick, N. Lape, S. Leary, K. Manville, E. Merkhofer, L. Strickhart, and M. Walmer. Apricot: A dataset of physical adversarial attacks on object detection. *https://arxiv.org/abs/1912.08166*, 2020.

[6] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[7] T. J. Chua, W. Yu, C. Liu, and J. Zhao. Detection of uncertainty in exceedance of threshold (duet): An adversarial patch localizer. In *IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, 2022.

[8] Alibaba Cloud. Qwen-vl: A vision-language model from alibaba cloud, 2023. URL `https://huggingface.co/Qwen/Qwen-VL`.

[9] Alibaba Cloud. Qwen2.5-vl-72b-instruct: A large multimodal model, 2024. URL `https://huggingface.co/Qwen/Qwen2.5-VL-72B-Instruct`.

[10] Google DeepMind. Gemini: Google's most capable ai model, 2024. URL `https://deepmind.google/technologies/gemini/`.

[11] B. Deng, D. Zhang, F. Dong, J. Zhang, M. Shafiq, and Z. Gu. Rust-style patch: A physical and naturalistic camouflage attacks on object detector for remote sensing images. *Remote Sensing*, 2023.

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

[13] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint*, arXiv:1608.00853, 2016.

[14] Yonggan Fu. Patch-fool: Are vision transformers always robust against adversarial perturbations?. *arXiv*, 2022.

[15] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[17] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*, 2023.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] R. H. Hwang, J. Y. Lin, S. Y. Hsieh, H. Y. Lin, and C. L. Lin. Adversarial patch attacks on deep-learning-based face recognition systems using generative adversarial networks. *Sensors*, 2023.

[20] C. Kang, Y. Dong, Z. Wang, S. Ruan, H. Su, and X. Wei. Diffender: Diffusion-based adversarial defense against patch attacks. *arXiv preprint arXiv:2306.09124*, 2024.

[21] Roie Kazoom, Raz Birman, and Ofer Hadar. Meta classification model of surface appearance for small dataset using parallel processing. *Electronics*, 11(21):3426, 2022.

[22] T. Kim, Y. Yu, and Y. M. Ro. Defending physical adversarial attack on object detection via adversarial patch-feature energy. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[24] Raz Lapid and Moshe Sipper. I see dead people: Gray-box adversarial attack on image-to-text models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 277–289. Springer, 2023.

[25] Raz Lapid, Zvika Haramaty, and Moshe Sipper. An evolutionary, gradient-free, query-efficient, black-box algorithm for generating adversarial instances in deep convolutional neural networks. *Algorithms*, 15(11):407, 2022.

[26] Raz Lapid, Almog Dubin, and Moshe Sipper. Fortify the guardian, not the treasure: Resilient adversarial detectors. *Mathematics*, 12(22):3451, 2024.

[27] Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black-box jailbreaking of large language models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL `https://openreview.net/forum?id=0SuyNOncxX`.

[28] Raz Lapid, Eylon Mizrahi, and Moshe Sipper. Patch of invisibility: Naturalistic black-box adversarial attacks on object detectors. In *6th Workshop on Machine Learning for Cybersecurity, part of ECMLPKDD 2024*, 2024.

[29] J. Liang, R. Yi, J. Chen, Y. Nie, and H. Zhang. Securing autonomous vehicles visual perception: Adversarial patch attack and defense schemes with experimental validations. *IEEE Transactions on Intelligent Vehicles*, 2024.

[30] S. Y. Lin, E. Chu, C. H. Lin, J. C. Chen, and J. C. Wang. Diffusion to confusion: Naturalistic adversarial patch generation based on diffusion model for object detector. *arXiv preprint arXiv:2307.08076*, 2023.

[31] J. Liu, A. Levine, C. H. L. Lau, R. Chellappa, and S. Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14973–14982, 2022.

[32] T. Liu, C. Yang, X. Liu, R. Han, and J. Ma. Rpau: Fooling the eyes of uavs via physical adversarial patches. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[33] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[35] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[36] Eylon Mizrahi, Raz Lapid, and Moshe Sipper. Pulling back the curtain: Unsupervised adversarial detection via contrastive auxiliary networks. *arXiv preprint arXiv:2502.09110*, 2025.

[37] Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

[38] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[39] Ben Pinhasov, Raz Lapid, Rony Ohayon, Moshe Sipper, and Yehudit Aperstein. XAI-based detection of adversarial attacks on deepfake detectors. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=7pBKrcn199`.

[40] Maura Pintor. Imagenet-patch: A dataset for benchmarking machine learning robustness against adversarial patches. *Pattern Recognition*, 2023.

[41] 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.

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[44] ByteDance Research. Ui-tars-72b-dpo: A 72b parameter vision-language model for ui understanding. `https://huggingface.co/bytedance-research/UI-TARS-72B-DPO`, 2024. Accessed: 2024-04-01.

[45] Kazoom Roie, Raz Birman, and Ofer Hadar. Improving the robustness of object detection and classification ai models against adversarial patch attacks. *arXiv preprint arXiv:2403.12988*, 2024.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[48] Angelo Sotgiu, Ambra Demontis, Marco Melis, Battista Biggio, Giorgio Fumera, Xiaoyi Feng, and Fabio Roli. Deep neural rejection against adversarial examples. *EURASIP Journal on Information Security*, 2020:1–10, 2020.

[49] C Szegedy. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[50] Snir Vitrack Tamam, Raz Lapid, and Moshe Sipper. Foiling explanations in deep neural networks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=wvLQMHtyLk.

[51] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[52] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[53] H. Wei, H. Tang, X. Jia, Z. Wang, H. Yu, Z. Li, S. Satoh, L. Van Gool, and Z. Wang. Physical adversarial attack meets computer vision: A decade survey. *arXiv preprint arXiv:2209.15179*, 2022.

[54] Xingxing Wei, Yao Huang, Yitong Sun, and Jie Yu. Unified adversarial patch for cross-modal attacks in the physical world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4445–4454, 2023.

[55] Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. {PatchCleanser}: Certifiably robust defense against adversarial patches for any image classifier. In *31st USENIX security symposium (USENIX Security 22)*, pages 2065–2082, 2022.

[56] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[57] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint*, arXiv:1704.01155, 2017.

[58] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society, 2018.

A Appendix: Ablation Study

We perform all evaluations on the ImageNet-Patch [40] dataset.

A.1 Effect of Parallelization

Parallelization significantly improves the efficiency of adversarial patch database creation. Since the application of each patch to each image-and the subsequent embedding computation-are independent operations, the process can be parallelized across multiple workers [21]. This enables rapid generation and encoding of large-scale patched image datasets.

In our setup, we applied adversarial patches to a collection of clean images, using a key-value approach where each image was divided into a 5×5 grid. Patch embeddings served as keys, while embeddings of image regions acted as values for retrieval. The end result was a database of 3,500 patch-image pairs with corresponding embeddings. To evaluate scalability, we measured execution time with varying levels of parallelism, confirming substantial speedups as the number of workers increased.

As shown in Table 3, using a single worker resulted in an average execution time of 24.57 minutes, whereas increasing the number of workers to six reduced the execution time to 3.58 minutes, demonstrating a 6.86× speedup. The results indicate that distributing the workload across multiple processes significantly reduces execution time while maintaining detection accuracy.

These findings validate the effectiveness of parallelization in our method, allowing it to scale efficiently for larger datasets. The speedup enables the rapid processing of extensive adversarial patch collections, making real-time detection feasible.

Table 3: Execution time for adversarial patch detection with different numbers of workers. Results are reported as mean ± standard deviation, in minutes.

Number of Workers	Execution Time (min)
1	24.57 ± 0.07
2	12.12 ± 0.10
3	8.11 ± 0.16
4	6.14 ± 0.26
5	4.59 ± 0.40
6	3.58 ± 0.54

A.2 Embedding Distance Analysis

We evaluate the effectiveness of our retrieval mechanism through an ablation study comparing several distance metrics for nearest-neighbor retrieval, including cosine similarity, L1 distance, L2 distance, and Wasserstein distance. All experiments in this subsection were conducted on the ImageNet-Patch dataset. Rather than relying solely on cosine similarity for retrieving stored adversarial patches, we also assess alternative metrics using embeddings extracted via CLIP [41].

Given an input image I, we partition it into grid-based regions and extract feature embeddings using CLIP's image encoder:

$$E_I = f(I), \quad E_{\mathcal{D}} = \{f(D_i) \mid D_i \in \mathcal{D}\}, \tag{9}$$

where $f(\cdot)$ denotes the CLIP embedding function and $\mathcal{D}$ represents the precomputed adversarial patch database.

For cosine similarity-based retrieval, the similarity score is computed as:

$$S(E_I, E_{\mathcal{D}}) = \frac{E_I \cdot E_{\mathcal{D}}}{\|E_I\| \, \|E_{\mathcal{D}}\|}, \tag{10}$$

with a stored adversarial patch retrieved if $S(E_I, E_{\mathcal{D}})$ exceeds a similarity threshold τ_s.

We also evaluate L1 and L2 distances. The L1 distance is defined as:

$$d_{\text{L1}}(E_I, E_{\mathcal{D}}) = \sum |E_I - E_{\mathcal{D}}|, \tag{11}$$

and the L2 distance is given by:

$$d_{\text{L2}}(E_I, E_{\mathcal{D}}) = \|E_I - E_{\mathcal{D}}\|_2. \tag{12}$$

For both L1 and L2 distances, retrieval is triggered when the computed distance falls below a threshold (τ_{L1} or τ_{L2}, respectively).

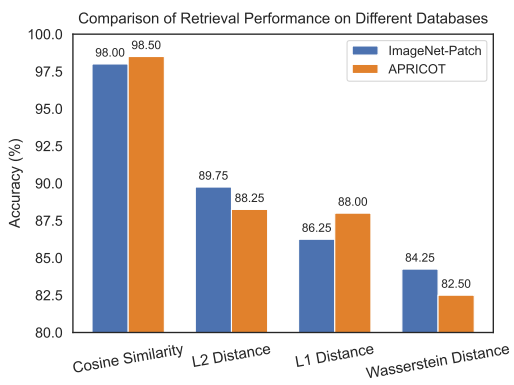
Additionally, we examine the Wasserstein distance, which measures the optimal transport cost between distributions. For two distributions P and Q over the embedding space, the Wasserstein distance is defined as:

$$W(E_I, E_{\mathcal{D}}) = \inf_{\gamma} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|], \tag{13}$$

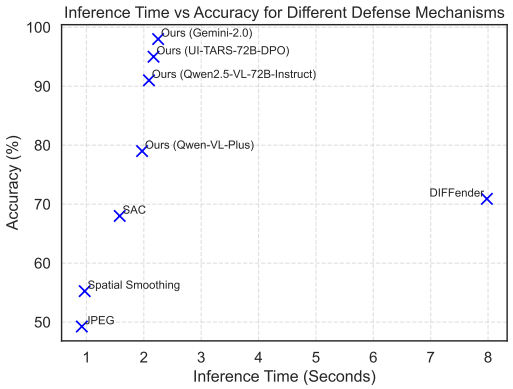
where γ is a joint distribution with marginals P and Q. This metric quantifies the minimal effort required to transport mass between the two embedding distributions.

We compare the retrieval effectiveness of these four metrics using Gemini-2.0 [10] for final classification. The cosine similarity-based approach achieves the highest classification accuracy at 98.00%, followed by L2 distance (89.75%), L1 distance (86.25%), and Wasserstein distance (84.25%). These results are visualized in Figure 3a.

These results indicate that cosine similarity most effectively captures the high-dimensional semantic relationships essential for robust adversarial patch retrieval, while the alternative metrics, although reasonable, perform less effectively-particularly the Wasserstein distance, which struggles to model distributional similarity from limited embedding samples.



(a) Retrieval performance using different distance metrics on CLIP embeddings.



(b) Inference time vs. accuracy for different defense mechanisms.

A.3 Inference Time Analysis

All experiments in this subsection were conducted on the ImageNet-Patch dataset. In addition to detection performance, we assess the inference time required for each defense mechanism. For an input image I, the processing time for a defense mechanism D is defined as:

$$T_D = \frac{1}{N} \sum_{i=1}^{N} t_i, \tag{14}$$

where t_i is the processing time for the i-th image and N is the total number of test images.

We analyze the trade-off between inference time T_D and classification accuracy A_D, which is calculated as:

$$A_D = \frac{C_{\text{correct}}}{C_{\text{total}}} \times 100, \tag{15}$$

with C_{correct} representing the number of correctly classified images and C_{total} the total number of images.

As shown in Figure 3b, JPEG compression [13] and Spatial Smoothing [57] offer the fastest inference times (0.92s and 0.97s, respectively), albeit with limited accuracy improvements (49.25% and 55.25%). SAC [31] requires 1.58s while achieving an accuracy of 68.00%, and DIFFender [20] attains an accuracy of 70.90% with an inference time of 7.98s.

Our method, leveraging Qwen-VL-Plus [8], Qwen2.5-VL-72B-Instruct [9], UI-TARS-72B-DPO [44], and Gemini-2.0 [10], achieves superior classification accuracy (79.00%, 91.00%, 95.00%, and 98.00%, respectively) with inference times of 1.97s, 2.09s, 2.17s, and 2.25s.

These findings highlight a clear performance–efficiency trade-off: higher detection accuracy generally demands increased computational cost. Our approach effectively balances these aspects by leveraging retrieval-augmented detection while maintaining inference times that remain competitive with existing defense mechanisms.

A.4 Prompt Engineering Analysis

All experiments in this subsection were conducted on the ImageNet-Patch dataset. To investigate the impact of prompt design [17] on adversarial patch detection, we conducted an ablation study evaluating five distinct prompting strategies. Each strategy aims to guide the VLM in classifying whether an image contains an adversarial patch. Given an input image I, the VLM is provided with a textual prompt $\mathcal{T}$ and returns a classification response:

$$R = \mathcal{V}(\mathcal{T}, I), \tag{16}$$

where $\mathcal{V}$ represents the VLM inference function.

To enhance context, we leverage a retrieved set of adversarial patch examples $\{P_1, \ldots, P_m\}$, where each P_i is an adversarial patch stored in the database, and a set of attacked images $\{I_1, \ldots, I_k\}$, where each I_j is a full image containing an applied adversarial patch. These elements provide additional visual references during inference.

The prompting strategies evaluated are as follows, along with the specific examples used:

1. **Instruction-only:** A generic instruction without examples:

   ```
   ‘‘Adversarial physical attacks involve placing random patches
   on images. You are an expert in identifying such patches.
   Is the following image attacked? Answer ’yes’ or ’no’.’’
   ```

2. **Attacked Images:** The instruction followed by examples of attacked images $\{I_1, \ldots, I_k\}$:

   ```
   ‘‘Here are examples of images that have been attacked: [Image
   1], [Image 2], [Image 3]. Given the next image, is it
   attacked? Answer ’yes’ or ’no’.’’
   ```

3. **Patch Examples:** The instruction followed by examples of extracted adversarial patches $\{P_1, \ldots, P_m\}$:

   ```
   ‘‘Here are examples of adversarial patches: [Patch 1], [Patch
   2], [Patch 3]. Given the next image, is it attacked? Answer
   ’yes’ or ’no’.’’
   ```

4. **Chain-of-Thought (CoT):** The instruction augmented with reasoning:

   ```
   ‘‘Adversarial attacks often involve adding suspicious patches.
   First, analyze if there are irregular regions. Then, decide
   if an attack is present. Is the following image attacked?
   Answer ’yes’ or ’no’.’’
   ```

5. **Combined (Final, Without CoT):** The instruction with both attacked images and patch examples:

   ```
   ‘‘Adversarial physical attacks involve random patches on
   images. You are an expert at detecting them. Here are
   examples of adversarial patches: [Patch 1], [Patch 2].
   Here are examples of attacked images: [Image 1], [Image
   2]. Given the above context, is this image attacked? Please
   answer ’yes’ or ’no’.’’
   ```

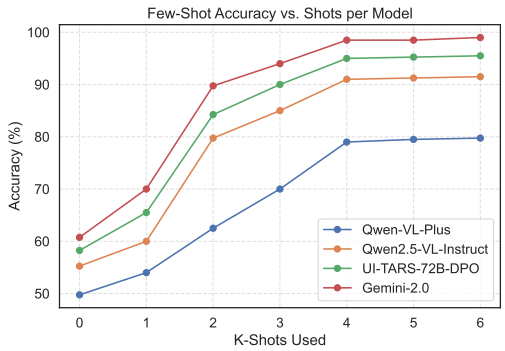
To quantify the effectiveness of each prompt type, we measured the detection accuracy $A_{\mathcal{T}}$ obtained under each configuration. The final selected prompt, as presented in 2, corresponds to the Combined (Final) strategy, which achieved the highest detection accuracy of 98.00%. The complete results are summarized in Figure 4b, where we observe that simple instructional prompts result in low accuracy (58.00%), while adding contextual examples (patches and attacked images) significantly improves performance. The CoT-based prompt further enhances accuracy to 91.25%, whereas the combined strategy achieves the highest overall detection rate.

This ablation study highlights that careful prompt engineering, particularly including few-shot visual examples and reasoning, is critical for maximizing VLM-based adversarial patch detection.

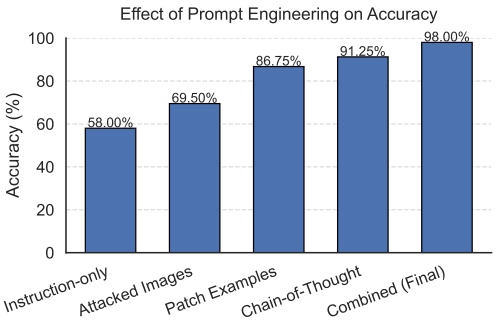
A.5 Impact of Few-Shot Context Size on Classification Accuracy

All experiments in this subsection were conducted on the ImageNet-Patch dataset. To evaluate the effect of context size on adversarial patch detection, we conducted an ablation study by varying the number of few-shot examples provided to the VLM during inference. Let $k \in \{0, 1, \ldots, 6\}$ denote the number of retrieved examples (i.e., the few-shot shots). For each k-shot configuration, we measured the classification accuracy A_k of the VLM in detecting adversarial patches across four different models: Qwen-VL-Plus, Qwen2.5-VL-Instruct, UI-TARS-72B-DPO, and Gemini-2.0.

Figure 4a illustrates the trend of A_k as a function of k. Across all models, we observe a consistent improvement in detection accuracy with increasing values of k, indicating that providing more contextual examples strengthens the model's ability to generalize and distinguish adversarial patterns.



(a) Few-shot detection accuracy across varying context sizes k.



(b) Effect of prompt engineering on adversarial patch classification accuracy.

Notably, UI-TARS-72B-DPO consistently achieves intermediate performance, surpassing Qwen-based models and closely approaching Gemini-2.0 accuracy.

These results suggest that larger few-shot contexts allow the VLM to better align the input query with prior adversarial patterns stored in the retrieval database. However, the performance gains tend to plateau beyond $k = 4$, highlighting a saturation effect where additional examples yield diminishing returns. The comparison also reveals that more capable VLMs (e.g., Gemini-2.0 and UI-TARS-72B-DPO) benefit more rapidly from few-shot conditioning than smaller models such as Qwen-VL-Plus and Qwen2.5-VL-Instruct, although Gemini-2.0 still demonstrates superior performance overall.

A.6 Generalization to Diverse Patch Shapes

Real-world adversarial patches appear in many shapes and textures, from geometric (square, round, triangular) to naturalistic or camouflage-like forms. To ensure robustness against these diverse patterns, we incorporate a range of patch types in the database creation phase. Concretely, each patch $P_i \in \mathcal{P}$ may be:

$$\text{square},\ \text{round},\ \text{triangle},\ \text{realistic}, \ldots$$

Since detection relies on embedding-based similarity rather than geometric assumptions, unusual or irregular patch shapes remain identifiable as long as their embeddings lie above a retrieval threshold τ. In practice, this approach allows our VRAG-based framework to detect both canonical patches and highly unobtrusive, adaptive adversarial artifacts designed to evade simpler defenses.

By collectively leveraging a rich database of patch embeddings, a retrieval-augmented paradigm, and a capable vision-language model, our method achieves robust generalization in adversarial patch detection across a wide spectrum of attack strategies.

A.7 Qualitative Results

In addition to quantitative evaluations, we present qualitative results highlighting the effectiveness of our proposed framework compared to existing defenses. Figure 6 illustrates visual comparisons across various defense mechanisms: Undefended, JPEG compression [13], Spatial Smoothing [57], SAC [31], DIFFender [20], and our method.

Adversarial patches remain clearly visible and disruptive in both Undefended and JPEG-compressed images, indicating that these methods fail to mitigate patch attacks effectively. SAC partially reduces the visibility of adversarial patches but does not consistently eliminate them, often leaving residual disruptions. DIFFender [20] demonstrates improved effectiveness compared to SAC by further reducing patch visibility, though residual disturbances remain apparent.

In contrast, our method reliably identifies and neutralizes adversarial patches, effectively mitigating their influence while preserving image integrity. However, our approach also has specific failure modes, particularly evident when the adversarial patch blends seamlessly into the noisy background of an image, matching its distribution. In such challenging cases (e.g., the last row of the right-hand

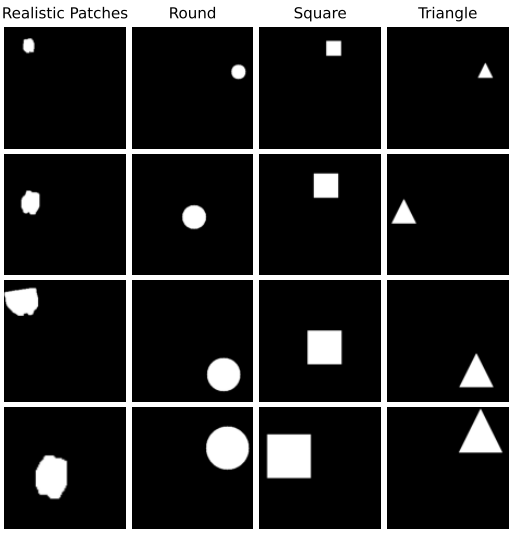


Figure 5: Examples of adversarial patch masks used in our dataset. We consider four types: realistic, round, square, and triangle. This diversity improves robustness across patch shapes.

table in Figure 6), the model may struggle to accurately differentiate between patch and background noise, highlighting a limitation to be addressed in future research.

A.8 Impact of Few-Shot Retrieval on VLM Accuracy

To further understand performance across different vision-language models (VLMs), Figure 7 shows confusion matrices for Qwen-VL-Plus [3], Qwen2.5-VL-72B [4], UI-TARS-72B-DPO [44], and Gemini-2.0 [52] under 0-shot, 2-shot, and 4-shot configurations. Increasing the number of retrieved examples consistently improves both true-positive and true-negative rates. Notably, the 4-shot configuration with Gemini-2.0 yields near-perfect separation between adversarial and clean samples. While Gemini-2.0 remains the top-performing model, UI-TARS-72B-DPO achieves highly competitive results, outperforming all other open-source VLMs by a significant margin.

These findings highlight the power of retrieval-augmented prompting for adversarial patch detection—especially when representative visual-textual context is injected via advanced VLMs.

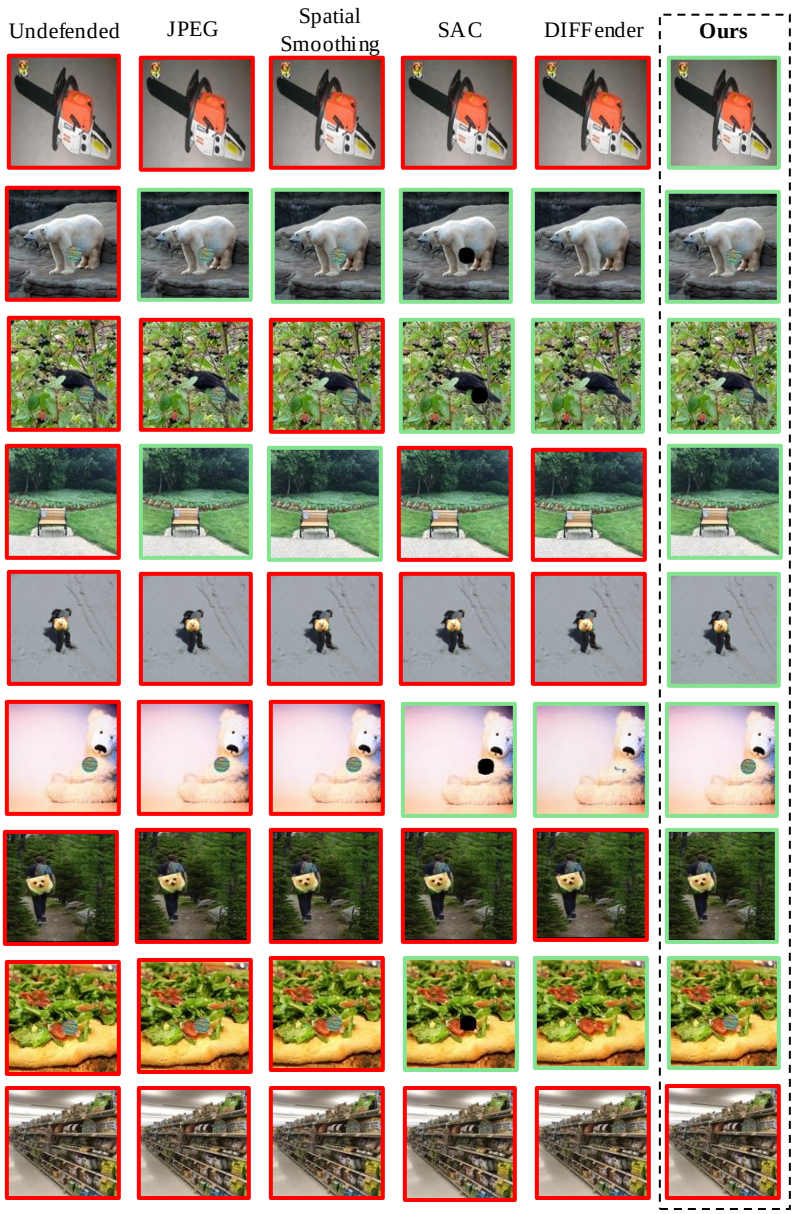


Figure 6: Qualitative comparison of different defense mechanisms. From left to right: Undefended, JPEG compression [13], Spatial Smoothing [57], SAC [31], DIFFender [20] and our method.

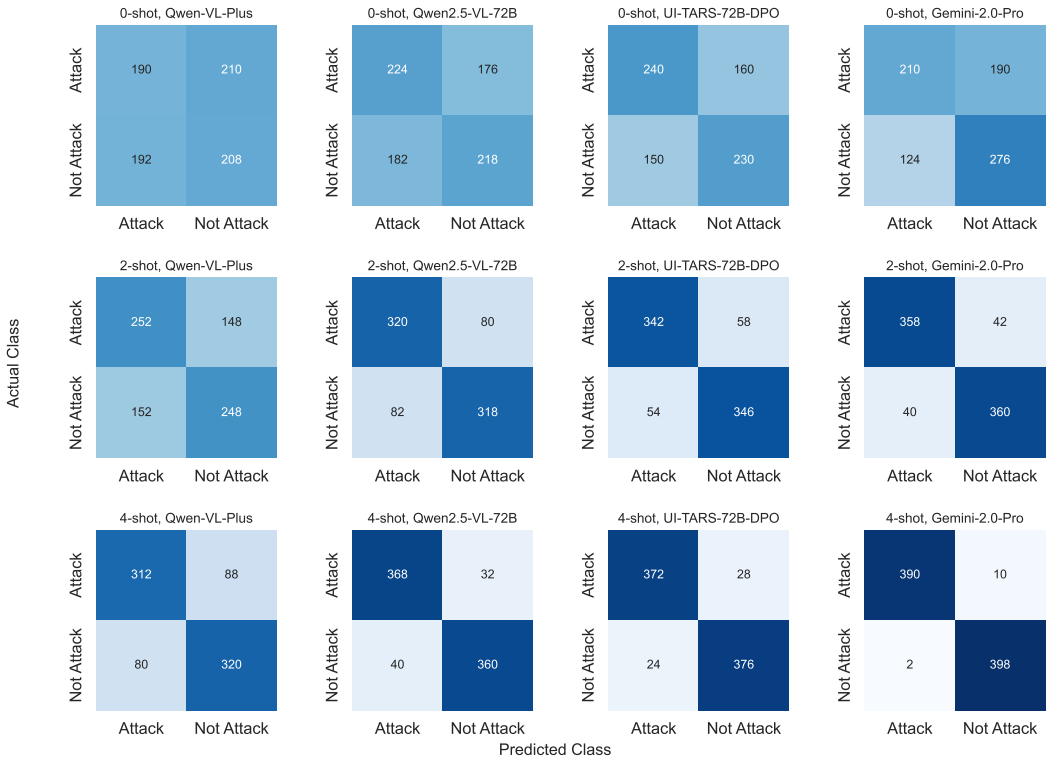


Figure 7: Confusion matrices across three few-shot configurations (rows) and four VLMs (columns). Axes represent predicted and actual classes ("Attack" vs. "Not Attack"). Gemini-2.0 achieves the best overall accuracy, while UI-TARS-72B-DPO offers the strongest open-source performance.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: The papers not including the checklist will be desk rejected. The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer **[Yes]**, **[No]**, or **[NA]**.
- **[NA]** means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "**[Yes]**" is generally preferable to "**[No]**", it is perfectly acceptable to answer "**[No]**" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "**[No]**" or "**[NA]**" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer **[Yes]** to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist",
- Keep the checklist subsection headings, questions/answers and guidelines below.
- Do not modify the questions and only use the provided macros for your answers.

1. Claims

1. **Question:** Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: [Yes] The abstract and introduction clearly state the paper’s main contributions-introducing don’t lag, rag, where we present a training free adversarial patch detector which outperform state of the art.

   Guidelines:

   - The answer NA 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 NA 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

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

Answer: [Yes] The Appendix (Limitation section) outlines key limitations, including reliance on synthetic scenes, limited dataset scale, and restricted reasoning and object diversity.

Guidelines:

- The answer NA 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

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

   Answer: [NA] The paper contains no theoretical results or proofs; Sections 3–6 focus solely on empirical methods and evaluations, instead, it focuses on dataset construction, evaluation protocols, and empirical findings.

   Guidelines:

   - The answer NA 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

1. **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] Sections 3.1 and 3.3, along with the Appendix, include detailed descriptions of dataset construction, evaluation setup, and all prompt templates, providing enough information for others to reproduce the benchmark and main experimental results.

Guidelines:

- The answer NA 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

1. **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] Section 3.1 and the Appendix describe that the PersianClevr dataset and scripts will be released with instructions to reproduce the benchmark and experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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

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

   Answer: [Yes] Sections 3.3 and 4.1–4.5 clearly specify dataset splits, evaluation settings, prompting strategies, and model usage, providing sufficient detail to understand the experimental setup.

   Guidelines:

   - The answer NA 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

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

   Answer: [Yes] All results provided and reproducible in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should provide error bars for all statistical results.
   - For plots, the authors should clearly state what the error bars represent (e.g., standard deviation, standard error of the mean, 95% confidence interval, etc.).

8. Experiments compute resources

1. **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] Appendix A.8 specifies GPU type, memory, cloud provider, and total compute time used, providing sufficient reproducibility details.

   Guidelines:

   - The answer NA 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

1. **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] The work follows the NeurIPS Code of Ethics, using only synthetic and publicly available data with no human subjects, privacy risks, or potential for harmful use.

Guidelines:

- The answer NA 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

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

   Answer: [NA] The work introduces a synthetic benchmark, which has no direct societal impact beyond methodological research.

   Guidelines:

   - The answer NA means that there is no societal impact of the work performed.
   - If the authors answer NA 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

1. **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., pretrained language models, image generators, or scraped datasets)?

   Answer: [NA] The paper poses no misuse risks, as it releases only synthetic benchmark data without real images, human content, or pretrained models requiring safeguards.

   Guidelines:

   - The answer NA 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

1. **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: [NA] The paper doesn't provide any external code/dataset.

   Guidelines:

   - The answer NA 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

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

   Answer: [NA] The paper doesn't provide any external code/dataset.

   Guidelines:

   - The answer NA 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

1. **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: [NA] The paper does not involve any crowdsourcing or research with human subjects; all data are synthetically generated and automatically processed.

   Guidelines:

   - The answer NA 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

1. **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: [NA] The research does not involve human subjects or participant studies, so no IRB or equivalent ethical review was required.

   Guidelines:

   - The answer NA 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

1. **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 rigorousness, or originality of the research, declaration is not required.

   Answer: [NA] The paper does not involve LLMs as any important, original, or non-standard components.

   Guidelines:

   - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.