

Article

Retrieval-Augmented Few-Shot Malware Detection via Binary Visualization and Vision–Language Embeddings

Woo Jin Jung [†], Nae-Joung Kwak ^{*}  and Byoung-Yup Lee ^{*} 

Department of Information Security, Pai Chai University, Daejeon 35345, Republic of Korea; 2284057@pcu.ac.kr or jungwoojin@korea.ac.kr

^{*} Correspondence: knj0125@pcu.ac.kr (N.-J.K.); bylee@pcu.ac.kr (B.-Y.L.)

[†] Current address: School of Cybersecurity, Korea University, Seoul 02841, Republic of Korea.

Abstract

The rapid evolution of malware families poses significant challenges for cybersecurity systems, particularly when newly emerging threats lack sufficient labeled data. Although image-based deep learning approaches have achieved strong performance under fully supervised conditions, their dependence on retraining limits adaptability in dynamic environments. To address this issue, we propose a Retrieval-Augmented Few-Shot Malware Detection Framework that integrates binary-to-image visualization, multimodal embedding using a frozen Vision–Language Model (Qwen2.5-VL), and similarity-based external memory retrieval. Malware binaries are converted into grayscale images and embedded into a semantic vector space without task-specific fine-tuning. During inference, query samples retrieve similar support embeddings from a vector database, and predictions are generated through similarity-weighted aggregation, enabling adaptation without parameter updates. Evaluated on the Mallmg dataset with 25 malware families under 1-shot to 10-shot settings, the framework achieves 0.886 accuracy in the 10-shot configuration. Ablation results demonstrate that combining VLM embeddings with retrieval mechanisms provides consistent improvements over individual components. These findings highlight the effectiveness of decoupling representation learning from adaptation for scalable few-shot malware detection.

Keywords: malware detection; few-shot learning; vision–language models; retrieval-augmented learning; binary-to-image visualization; external memory; Qwen2.5-VL



Academic Editors: Jian Xu
and Ruijin Wang

Received: 6 March 2026

Revised: 16 April 2026

Accepted: 19 April 2026

Published: 22 April 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The rapid proliferation of malware continues to pose significant challenges to global cybersecurity infrastructures, with new variants and families emerging at an unprecedented rate [1]. Traditional signature-based detection techniques remain effective for known threats; however, they exhibit fundamental limitations against modern evasion strategies such as polymorphism, metamorphism, packing, encryption, and code obfuscation [2]. Malware families continuously evolve through shared code lineage, command-and-control infrastructures, and behavioral patterns, making family-level classification a core task in threat intelligence and incident response [3]. In real-world operational environments, however, newly observed malware families often lack sufficient labeled samples, leading to severe data scarcity that directly degrades detection performance and delays defensive actions [4].

To address these limitations, recent studies have proposed deep learning-based approaches that transform malware binaries into visual representations and apply convolutional neural networks (CNNs) for classification [5,6]. Binary-to-image conversion exposes structural characteristics of executables as texture-like patterns, enabling strong classification performance under fully supervised settings. Nevertheless, such approaches rely heavily on large labeled datasets and require retraining or fine-tuning whenever new families emerge. This requirement is impractical in rapidly evolving threat landscapes [4,6]. Consequently, malware detection faces a few-shot learning (FSL) problem, where models must generalize to previously unseen families using only a small number of labeled samples [7].

Few-shot learning has been extensively studied under metric-based, optimization-based, and memory-augmented paradigms. Representative methods include Prototypical Networks [8], which classify queries based on distances to class prototypes, and Model-Agnostic Meta-Learning (MAML) [9], which aims to rapidly adapt model parameters to new tasks with minimal gradient updates. In cybersecurity domains, few-shot frameworks have been applied to malware visualization [10], malware variant graphs [11], and network intrusion detection [12], demonstrating that limited supervision can still enable recognition of novel threats. However, most existing approaches rely on single-modality representations (e.g., images or API sequences) or fixed meta-learning architectures, which limits their scalability when encountering heterogeneous malware patterns or extremely sparse support sets.

Meanwhile, recent advances in Vision-Language Models (VLMs) have introduced new opportunities for representation learning beyond conventional CNN-based pipelines. Large-scale multimodal pretraining aligns visual inputs with semantic embedding spaces, enabling strong generalization even under limited downstream supervision [13]. At the same time, Retrieval-Augmented Generation (RAG) frameworks have demonstrated that integrating external memory with similarity-based retrieval can improve performance in data-scarce settings by dynamically incorporating relevant examples during inference [14,15]. Despite these advances, the integration of VLM-based multimodal embeddings with retrieval-based external memory mechanisms for few-shot malware detection has not been sufficiently explored.

In this paper, we propose a Retrieval-Augmented Few-Shot Malware Detection Framework that integrates binary-to-image visualization, multimodal embedding via Qwen2.5-VL, and external vector memory. Unlike prior few-shot malware studies that depend on meta-learning classifiers or fixed prototype structures [8–11], our approach adopts a retrieval-centric paradigm. Malware binaries are first converted into grayscale images, embedded into a semantic vector space using a frozen VLM encoder, and stored in an external vector database. During inference, query samples retrieve the most similar support embeddings via cosine similarity, and final family predictions are derived through similarity-weighted aggregation. This design enables dynamic adaptation to newly emerging malware families without task-specific retraining.

While frozen embeddings combined with nearest-neighbor retrieval have been explored in prior studies, their direct application to malware detection remains non-trivial due to the unique structural and distributional properties of malware data. Unlike conventional retrieval-based frameworks that rely on generic feature spaces, the proposed method explicitly adapts similarity-based reasoning to malware structural characteristics, where intra-family variability and obfuscation-induced distortions are prevalent. In particular, this study combines multimodal pre-trained representation learning with an external memory retrieval mechanism to propose a structural adaptation strategy distinct from conventional meta-learning-based approaches. By shifting adaptation from parameter

updates to retrieval-based scalability, the proposed framework provides a novel perspective for addressing few-shot malware detection in highly dynamic threat environments.

The main contributions of this work are summarized as follows.

1. We propose a retrieval-augmented few-shot malware detection framework that integrates binary-to-image visualization, multimodal Vision–Language Model (VLM) embeddings, and external vector memory. Unlike conventional few-shot learning approaches that rely on task-specific parameter updates, the proposed framework enables dynamic adaptation to newly emerging malware families without retraining by decoupling representation learning from inference.
2. We demonstrate that frozen Qwen2.5-VL embeddings provide semantically enriched and transferable representations for malware images. Even without task-specific fine-tuning, these embeddings significantly improve few-shot generalization performance compared to conventional vision-only models, highlighting the effectiveness of multimodal pretraining in cybersecurity applications.
3. We introduce a retrieval-centric classification mechanism based on cosine similarity and similarity-weighted aggregation. In contrast to standard nearest-neighbor or prototype-based methods, the proposed aggregation strategy is designed to account for structural variability in malware images, improving robustness against intra-family diversity and noisy support samples.
4. We conduct extensive experiments on the Mallmg dataset under multiple N-way K-shot settings, along with cross-dataset evaluations and ablation studies. The results demonstrate that the proposed framework achieves competitive performance with fully supervised models while maintaining strong adaptability in low-data scenarios.

The remainder of this paper is organized as follows. Section 2 reviews related work on malware analysis, deep learning-based detection, and few-shot learning. Section 3 describes the proposed framework in detail, including malware visualization, VLM-based embedding, external memory construction, and retrieval-based classification. Section 4 presents experimental results and quantitative evaluations. Finally, Section 5 concludes the paper and discusses future research directions.

2. Literature Review

2.1. Malware Detection Approaches

Malware detection has been extensively studied, with approaches broadly categorized into static, dynamic, and hybrid analysis techniques. Traditional static analysis methods rely on signature matching and handcrafted features extracted from executable files [1,3]. While efficient, these approaches often struggle to detect obfuscated or previously unseen malware variants. Dynamic analysis, which monitors runtime behavior, provides more robust detection but incurs significant computational overhead and requires controlled execution environments [4].

To address these limitations, machine learning-based approaches have been introduced, leveraging large-scale datasets such as EMBER [5]. In particular, deep learning models have demonstrated strong performance by automatically learning discriminative features from raw binary data [7]. However, these methods typically require large amounts of labeled data, which are often unavailable for newly emerging malware families.

2.2. Image-Based Malware Representation

A notable advancement in malware analysis is the transformation of binary files into image representations [6]. This approach converts byte sequences into grayscale images, enabling the application of computer vision techniques for malware classification.

Subsequent studies have shown that convolutional neural networks (CNNs) can effectively capture structural patterns in malware images [8]. Image-based representations preserve spatial relationships within binary data, allowing models to distinguish between different malware families based on visual features.

These advances highlight the importance of robust feature extraction in malware detection tasks.

2.3. Few-Shot Malware Detection

Few-shot learning has emerged as a promising paradigm for addressing data scarcity in malware detection. Unlike conventional supervised learning, few-shot methods aim to generalize from a limited number of labeled samples.

Prototypical Networks (ProtoNet) [9] and Model-Agnostic Meta-Learning (MAML) [10] are representative approaches that have been widely applied to few-shot classification tasks. In the context of malware detection, recent studies have explored visualization-based methods [11] and graph-based contrastive learning approaches [12] to improve generalization across unseen malware families.

Despite these advancements, most few-shot learning methods require task-specific adaptation or retraining, which limits their applicability in dynamic cybersecurity environments where rapid response is essential.

2.4. Vision–Language Models and RAG

Retrieval-augmented learning has recently gained attention as a mechanism for enhancing model performance without retraining. These approaches combine representation learning with external memory retrieval to improve inference.

Early work on retrieval-augmented generation (RAG) demonstrated the effectiveness of incorporating external knowledge into neural models [16]. Similarly, nearest-neighbor-based methods have been shown to improve generalization by leveraging stored representations [17].

More recent studies have explored retrieval mechanisms in the context of in-context learning and prompt selection [18]. These methods highlight the potential of retrieval-based reasoning for improving adaptability in low-data scenarios.

2.5. Vision–Language Models and Recent Advances

Recent advances in Vision–Language Models (VLMs) have enabled powerful multimodal representation learning by aligning visual and textual information in a shared embedding space. Models such as CLIP [19], ALIGN [20], and ALBEF [21] have demonstrated strong generalization capabilities across a wide range of vision tasks.

In the context of malware analysis, representation learning approaches have also been explored. For example, end-to-end deep learning models that directly process binary data have demonstrated scalable malware detection performance [22]. Additionally, recent studies have investigated n-gram-based feature representations to improve classification accuracy by capturing local byte-level patterns [23].

These approaches are further supported by advances in self-supervised and contrastive learning, including SimCLR [24], MoCo [25], and BYOL [26], which enable effective feature learning without requiring large labeled datasets. Vision Transformer-based self-supervised methods have also demonstrated strong representation capabilities [27].

Recent studies have further explored the integration of vision–language models into few-shot learning frameworks. Applications of VLM-based few-shot learning have been reported across various domains, demonstrating strong generalization capability [28].

Training-free approaches using CLIP embeddings have demonstrated competitive performance in low-data scenarios [29].

In cybersecurity, knowledge-driven and multimodal approaches have been proposed for tasks such as ransomware detection using malware knowledge graphs [30].

Additionally, multimodal alignment strategies have been shown to improve few-shot generalization [31], while recent works on vision–language-based classification further confirm the effectiveness of these approaches [32].

Despite these advances, existing approaches often require retraining or adaptation when encountering new classes. In contrast, the proposed framework leverages VLM-based embeddings together with retrieval-augmented inference, enabling scalable and training-free malware classification in dynamic environments.

Table 1 provides a comprehensive overview of the related works discussed in Section 2, categorized by research direction. Traditional malware detection approaches rely on signature-based or supervised learning techniques, while recent studies focus on image-based representations and deep learning models. Few-shot learning methods address data scarcity by enabling generalization from limited samples, and retrieval-augmented approaches enhance inference through external memory. Recent advances in vision–language models and self-supervised learning further improve representation quality and generalization capability, forming the foundation of the proposed framework.

Table 1. Comprehensive summary of related works discussed in Section 2.

Refs.	Year	Category	Dataset/Domain	Methodology	Key Contribution
[1]	2020	Malware Detection	Various datasets	Signature-based & heuristic	Comprehensive review of traditional malware detection approaches
[3]	2010	Malware Detection	General	Obfuscation analysis	Early survey on malware obfuscation techniques
[4]	2012	Malware Detection	General	Dynamic analysis	Systematic survey of automated dynamic malware analysis techniques and tools
[5]	2018	Malware Dataset	PE files	EMBER dataset	Large-scale benchmark dataset for static malware detection
[7]	2015	Malware Detection	Binary programs	DNN-based feature extraction	Deep learning-based feature extraction from binary programs
[6]	2011	Image-based Malware	Mallmg	Binary-to-image conversion	Introduced malware visualization via grayscale image representation
[8]	2020	Image-based Malware	Mallmg	CNN ensemble-based classification	High-performance malware classification using ensemble CNN architectures
[9]	2017	Few-shot Learning	MiniImageNet	Prototypical Networks	Metric-based few-shot learning via class prototype representation
[10]	2017	Few-shot Learning	General meta-learning tasks	MAML	Gradient-based meta-learning enabling fast adaptation across tasks
[11]	2022	Few-shot Malware	Mallmg	Siamese Network	Few-shot malware classification using image similarity learning
[12]	2022	Few-shot Malware	Malware graphs	FewM-HGCL	Graph contrastive learning for malware variant detection
[16]	2020	Retrieval-Augmented Learning	NLP tasks	RAG	Retrieval-augmented generation combining parametric and non-parametric memory

Table 1. Cont.

Refs.	Year	Category	Dataset/Domain	Methodology	Key Contribution
[17]	2020	Retrieval-based Learning	Language modeling	kNN-based inference	Improves language model generalization via nearest-neighbor memorization
[18]	2022	Retrieval-based Learning	In-context learning	Prompt retrieval	Enhances in-context learning via retrieval-based prompt selection
[19]	2021	Vision–Language Models	General vision–language tasks	CLIP	Aligns vision and language representations using contrastive learning
[20]	2021	Vision–Language Models	Large-scale multimodal data	ALIGN	Scalable vision–language representation learning with noisy supervision
[21]	2021	Vision–Language Models	Vision–language tasks	ALBEF	Vision–language alignment with momentum distillation and fusion
[22]	2017	Malware Representation	Binary files	End-to-end deep learning	Direct malware classification from raw executable binaries
[23]	2023	Malware Representation	Malware binaries	n-gram modeling	Byte-level n-gram features for malware detection and classification
[24]	2020	Representation Learning	Vision tasks	SimCLR	Contrastive self-supervised representation learning framework
[25]	2020	Representation Learning	Vision tasks	MoCo	Momentum contrast for unsupervised representation learning
[26]	2020	Representation Learning	Vision tasks	BYOL	Self-supervised learning without negative samples
[27]	2021	Representation Learning	Vision transformers	DINO	Self-supervised representation learning for vision transformers
[28]	2025	VLM Few-shot	Agriculture (wheat grains)	Multimodal few-shot learning	Vision–language integration for domain-specific few-shot classification
[29]	2025	VLM Few-shot	General vision	CLIP + KDE	Training-free few-shot classification using CLIP embeddings and density estimation
[30]	2025	Cybersecurity AI	Ransomware detection	Knowledge graph + zero-shot learning	Transductive zero-shot ransomware detection using malware knowledge graphs
[31]	2024	VLM Few-shot	Multimodal tasks	VL-Few	Vision–language alignment for multimodal few-shot meta-learning
[32]	2024	VLM Few-shot	Agriculture (crop disease)	VLM-based classification	Domain-specific few-shot classification using vision–language models

3. Methodology

3.1. Framework Overview

Figure 1 illustrates the overall architecture of the proposed retrieval-augmented few-shot malware detection framework. The system consists of three main components: (i) a malware visualization module that converts executable binaries into structured grayscale images, (ii) a Vision–Language Model (VLM)-based feature extraction module that embeds malware

label enrichment. This design choice simplifies deployment while still benefiting from the multimodal pretraining of the model.

The advantage of using a Vision–Language Model over conventional vision-only models lies in the richness of its embedding space. Even when only the visual encoder is used, the representations are influenced by multimodal alignment learned during pretraining. As a result, the embeddings capture more informative and transferable features compared to vision-only architectures such as Vision Transformers (ViT) or self-supervised models such as DINOv2.

From an implementation perspective, all input images are resized to a fixed resolution of 384×384 and processed through a model-specific processor. The encoder produces 2048-dimensional embeddings, which are extracted using a weighted aggregation of the last four hidden layers. This pooling strategy improves representation stability and enhances similarity-based retrieval performance.

Furthermore, Qwen2.5-VL is deployed in a frozen configuration without task-specific fine-tuning. This design is particularly suitable for malware analysis scenarios, where retraining is costly and impractical in dynamic environments. By decoupling representation learning from downstream adaptation, the framework enables scalable few-shot inference through retrieval-based mechanisms.

The structural distinction between conventional CNN-based fine-tuning approaches and the proposed retrieval-centric paradigm is illustrated in Figure 2.

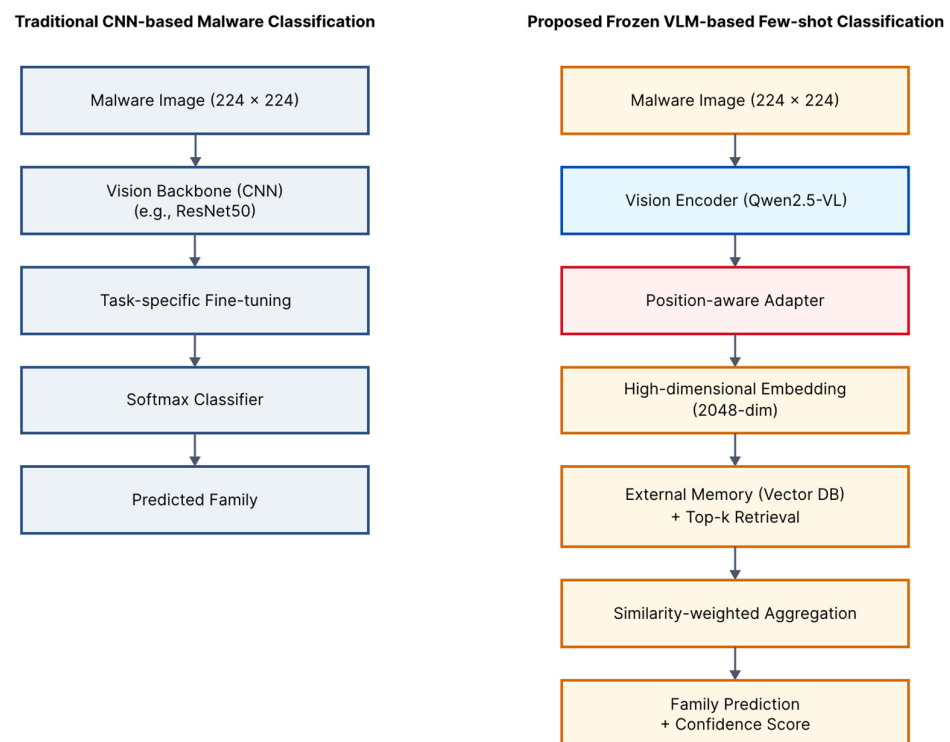


Figure 2. Architectural comparison between CNN fine-tuning and frozen VLM with retrieval-based adaptation.

As shown in Figure 2, traditional pipelines rely on task-specific parameter updates and softmax-based classification, whereas the proposed framework maintains a frozen encoder and performs adaptation through similarity-based retrieval over external memory.

This architectural separation between representation learning and adaptation enables flexible incorporation of new malware families without retraining, which is particularly advantageous in few-shot malware classification scenarios.

Overall, the use of Qwen2.5-VL allows the proposed framework to leverage semantically structured embeddings while maintaining a scalable and retrieval-driven adaptation strategy.

As shown in Figure 2, conventional CNN pipelines depend on parameter updates to accommodate new classes, whereas the proposed framework maintains a frozen encoder and performs task adaptation through similarity-based retrieval. This architectural separation between representation learning and adaptation is particularly advantageous in few-shot malware classification scenarios, where labeled data are limited and rapid deployment is required.

3.3. External Memory and Vector Database

Support samples are stored in an external memory implemented as a vector database. Each malware image is embedded using Qwen2.5-VL, producing a normalized feature vector of dimension 2048. These embeddings are indexed using FAISS (version 1.13.2, faiss-cpu) to enable efficient similarity-based retrieval.

Specifically, we employ a FAISS IndexFlatIP structure, which performs exact nearest neighbor search based on inner product similarity. To compute cosine similarity, all embedding vectors are L2-normalized prior to indexing. This allows cosine similarity to be efficiently computed using inner product operations while avoiding approximation errors introduced by clustering-based indexing methods.

Formally, let

$$\mathcal{S} = \{(z_i, y_i)\}_{i=1}^M \quad (1)$$

denote the support set, where z_i represents the embedding of the i -th malware sample and y_i is its corresponding family label. Each embedding is stored in the vector database together with its associated metadata.

Given a query embedding z_q , similarity scores are computed as:

$$\text{sim}(z_q, z_i) = z_q^\top z_i \quad (2)$$

using the inner product between the query and support embeddings. The system retrieves the top- k nearest neighbors based on these similarity scores.

$$\mathcal{N}_k(z_q) = \text{TopK}(z_q) \quad (3)$$

The Top- k retrieval mechanism plays a critical role in stabilizing classification by incorporating multiple relevant support samples, thereby reducing sensitivity to noise and individual sample variability. To analyze its impact, we conducted ablation experiments with $k \in \{1, 3, 5, 10\}$. The results show that increasing k improves robustness by aggregating information from multiple neighbors, with the best performance achieved at $k = 10$. Therefore, $k = 10$ is adopted as the default setting in the proposed framework.

This retrieval-based design enables dynamic incorporation of new malware families by simply appending additional embeddings to the database without requiring model retraining. Furthermore, the use of exact nearest neighbor search ensures stable and consistent retrieval performance, which is particularly important in few-shot classification scenarios where small variations in similarity can significantly affect predictions.

3.4. N-Way K-Shot Episodic Protocol

Evaluation is conducted using an N-way K-shot episodic protocol, which is widely adopted in few-shot learning scenarios. In each episode, N malware families are randomly selected from the dataset, and K labeled samples per class are used to construct the support

set. In addition, a fixed number of query samples (five per class) are independently sampled from the same classes to form the query set.

The overall episodic sampling process is illustrated in Figure 3. As shown in the figure, each episode consists of multiple malware families, where support and query samples are drawn independently for each class. The support set is used to populate the external memory, while the query set is used to evaluate classification performance within the same episode.

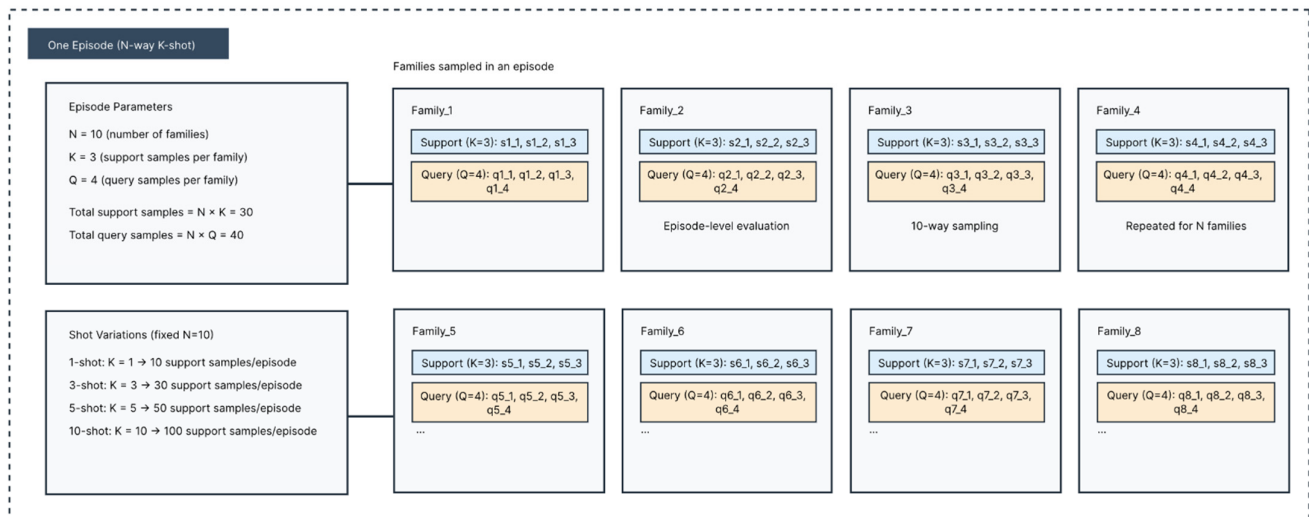


Figure 3. Episodic N-way K-shot setup used in the proposed few-shot malware detection framework.

Each episode is constructed as follows. First, N classes are randomly selected. Second, for each class, K support samples are drawn and stored in the external memory as embedding vectors. Third, query samples are independently sampled from the same classes. Finally, each query is processed through the retrieval-based classification pipeline described in Section 3.5.

To ensure reproducibility, a fixed random seed (42) is used for all sampling procedures. Furthermore, multiple episodes are generated, and the reported results correspond to the average performance across these episodes. This design reduces variance caused by random sampling and provides a more reliable evaluation of the model.

The episodic configuration in this study follows a 5-way classification setting with $K \in \{1, 3, 5, 10\}$. All baseline methods are evaluated under the same episodic setup to ensure fair comparison. This unified evaluation protocol allows consistent comparison between the proposed retrieval-based framework and conventional few-shot learning approaches.

3.5. Retrieval-Augmented Classification Algorithm

The proposed framework performs classification through a retrieval-augmented inference process, in which predictions are generated based on similarity relationships between query and support embeddings. The overall procedure is illustrated in Figure 4 and formally described in Algorithm 1.

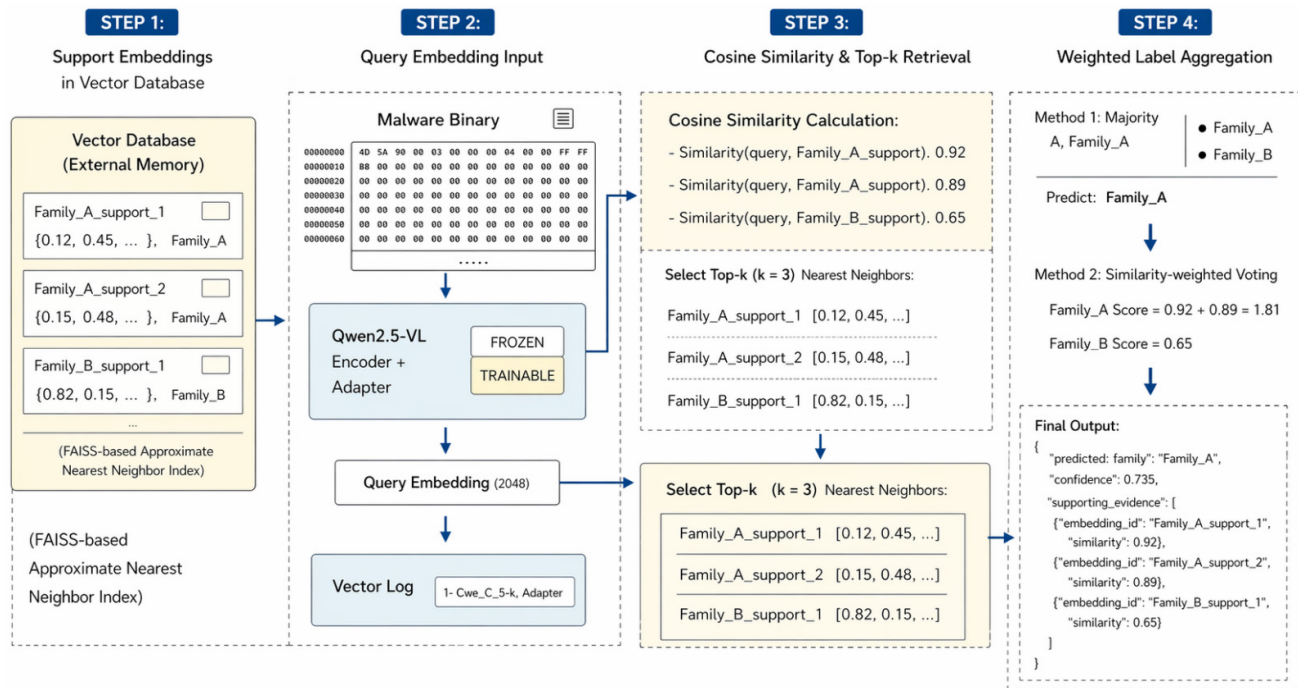


Figure 4. Retrieval-augmented similarity aggregation for few-shot malware classification.

As shown in Figure 4, the process consists of four main steps. First, support embeddings are precomputed and stored in an external memory implemented as a vector database. Second, a query malware sample is converted into an image and passed through the Qwen2.5-VL encoder to obtain a query embedding. Third, similarity scores between the query embedding and all stored support embeddings are computed, and the top-k nearest neighbors are retrieved. Finally, classification is performed using an aggregation strategy based on the retrieved neighbors.

Formally, given a query embedding z_q , the system retrieves the set of top-k nearest neighbors $\mathcal{N}_k(z_q)$ from the vector database. Each neighbor is associated with a label y_i and a similarity score $\text{sim}(z_q, z_i)$.

Instead of relying on simple majority voting, the proposed framework employs a similarity-weighted aggregation scheme. For each class c , a score is computed by summing the similarity values of retrieved neighbors belonging to that class:

$$\text{score}(c) = \sum_{(z_i, y_i) \in \mathcal{N}_k(z_q)} 1(y_i = c) \cdot \text{sim}(z_q, z_i) \quad (4)$$

where $1(\cdot)$ is an indicator function. The final prediction is obtained by selecting the class with the highest aggregated score:

$$\hat{y} = \arg \max_c \text{score}(c) \quad (5)$$

This procedure is summarized in Algorithm 1. Compared to conventional k-nearest neighbor classification, the similarity-weighted scheme allows the model to account for the relative importance of each retrieved neighbor. Highly similar samples contribute more to the final prediction, while less relevant neighbors have reduced influence. This improves robustness, particularly in scenarios where the retrieved neighbors include samples from multiple classes.

Algorithm 1. Retrieval-augmented few-shot classification procedureInput: Query image x_q , support embeddings $\{(z_i, y_i)\}$, k Output: Predicted family label $\hat{y}$ 1: $z_q \leftarrow \text{VLM_Encoder}(x_q)$ 2: Retrieve top- k neighbors $N_k(z_q)$ from vector database3: For each class c :

$$\text{score}(c) = \sum_{(z_i, y_i) \in N_k(z_q)} 1(y_i = c) \cdot \text{sim}(z_q, z_i) \quad \hat{y} = \arg \max_c \text{score}(c)$$

4: return $\hat{y}$

The retrieval-based classification design further enables transparent and interpretable decision-making. As illustrated in Figure 4, the system can provide supporting evidence in the form of retrieved neighbors and their similarity scores, allowing analysts to trace the reasoning behind each prediction.

4. Experimental Evaluation

4.1. Experimental Setup and Baselines Configuration

To evaluate the proposed retrieval-augmented few-shot malware detection framework, experiments were conducted on the MallImg dataset [6], which consists of grayscale malware images categorized into 25 families. All images were resized to 384×384 pixels prior to embedding.

Evaluation follows an N-way K-shot episodic protocol, which is widely adopted in few-shot learning research [8,9]. In this study, we use $N = 5$ families and $K \in \{1, 3, 5, 10\}$ labeled samples per family. For each episode, support embeddings are stored in external memory, and query samples are classified through retrieval-based similarity aggregation.

The episodic protocol and classification mechanism are described in Section 3.

All results are averaged over 5 independent episodes with different random samplings.

All experiments were conducted on a GPU-enabled system using mixed-precision inference. The Qwen2.5-VL encoder was deployed in a frozen configuration without task-specific fine-tuning. Embeddings were indexed using FAISS IndexFlatIP, enabling exact nearest neighbor search based on cosine similarity computed via L2-normalized embeddings.

To provide a comprehensive comparison, the proposed method was evaluated against several baseline models, including:

- A conventional CNN classifier trained from scratch,
- ResNet-18 fine-tuned on the target dataset,
- A Vision Transformer (ViT)-based classifier,
- Prototypical Networks (ProtoNet) [8],
- Model-Agnostic Meta-Learning (MAML) [9],
- A retrieval-only variant without similarity-weighted aggregation.

All baseline models were trained and evaluated under the same episodic setting to ensure fair comparison. For supervised baselines, standard cross-entropy loss and Adam optimization were used with identical data splits. Implementation details were aligned to maintain consistent evaluation conditions across models.

Performance was measured using Accuracy, Precision, Recall, and F1-score [33–35]. Accuracy evaluates overall classification correctness, while Precision and Recall assess class-level discrimination performance. The F1-score provides a balanced measure of Precision and Recall, which is particularly important in few-shot scenarios where class imbalance and limited support samples may affect performance [12].

All experiments were conducted on a workstation equipped with an NVIDIA RTX 4080 SUPER GPU (16 GB VRAM), Ryzen 9 9900X3D CPU, and 64 GB RAM. The implementation was based on PyTorch (2.7.1+cu118) with CUDA acceleration.

4.2. Quantitative Results

4.2.1. Few-Shot Performance Across Shot Settings

Table 2 summarizes the few-shot classification performance of the proposed framework under different shot configurations.

Table 2. Few-shot classification performance under different shot settings.

Shot	Accuracy	Precision	Recall	F1-Score
1-shot	0.878	0.836	0.775	0.793
3-shot	0.875	0.794	0.734	0.751
5-shot	0.864	0.807	0.736	0.757
10-shot	0.886	0.806	0.753	0.773

As shown in Table 2, the classification accuracy remains relatively stable across shot settings. The highest accuracy is observed at the 10-shot configuration (0.886), while 1-shot and 3-shot settings achieve comparable performance (0.878 and 0.875, respectively). Interestingly, the 5-shot setting yields slightly lower accuracy (0.864), indicating that performance does not monotonically increase with additional support samples.

This behavior suggests that increasing the number of support samples does not always guarantee improved discrimination. In episodic evaluation, additional support embeddings may introduce intra-class redundancy or sampling variability, which can slightly influence similarity-weighted aggregation outcomes. Nevertheless, the overall variation across shot settings is limited, demonstrating stable few-shot generalization.

In addition to accuracy, Precision, Recall, and F1-score follow a similar trend across shot configurations. The F1-score remains within a narrow range (0.751–0.793), indicating consistent class-level discrimination performance under varying support sizes.

Figure 5 illustrates the relationship between the number of shots and classification accuracy.

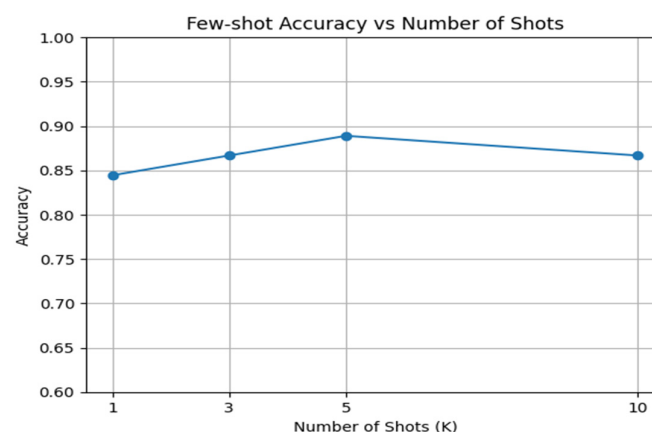


Figure 5. Few-shot accuracy as a function of the number of shots.

As shown in Figure 5, performance fluctuations across shot settings are minor, confirming that the proposed retrieval-augmented framework maintains robust behavior even with limited labeled samples.

4.2.2. Comparison with Fully Supervised Baselines

To assess competitiveness, we compare the proposed framework with conventional supervised classifiers. Table 3 summarizes the classification accuracy of the proposed few-shot framework and several fully supervised baseline models on the MallImg dataset.

Table 3. Performance comparison between the proposed few-shot framework and supervised baselines.

Method	Training Type	Accuracy
CNN (from scratch)	Supervised	0.641
ResNet-18 (fine-tuned)	Supervised	0.912
SVM	Supervised	0.914
Vision Transformer (ViT)	Supervised	0.818
Proposed (10-shot)	Few-shot (No retraining)	0.886

Table 3 shows that fully supervised models such as ResNet-18 (0.912) and SVM (0.914) achieve higher absolute accuracy. However, these methods require complete retraining with fully labeled datasets. In contrast, the proposed few-shot framework achieves competitive performance (0.886 at 10-shot) without updating encoder parameters. This highlights a critical advantage in dynamic malware environments, where rapid adaptation to newly emerging families is required.

Notably, the conventional CNN baseline (0.641) demonstrates substantially lower performance, indicating that training from scratch under limited supervision is insufficient for robust malware family classification.

4.2.3. Comparison with Few-Shot Learning Baselines

To further evaluate the effectiveness of the proposed framework in few-shot scenarios, we compare it with representative few-shot learning approaches, including Prototypical Networks (ProtoNet) and Model-Agnostic Meta-Learning (MAML). All models are evaluated under the same 5-way 10-shot episodic setting to ensure fair comparison.

Table 4 summarizes the classification performance of the proposed method and baseline few-shot models on the MallImg dataset.

Table 4. Performance comparison between the proposed few-shot framework and few-shot learning baselines.

Method	Training Type	Accuracy
Prototypical Networks (ProtoNet)	Few-shot	0.854
MAML	Few-shot	0.861
Proposed (10-shot)	Few-shot (No retraining)	0.886

As shown in Table 4, the proposed framework outperforms all compared few-shot learning baselines under the same evaluation setting. In particular, it achieves higher accuracy than Prototypical Networks (0.854) and MAML (0.861), demonstrating the effectiveness of retrieval-augmented inference combined with semantically rich embeddings.

Unlike ProtoNet, which relies on class prototype averaging, the proposed method aggregates multiple retrieved samples using similarity-weighted scoring, allowing it to better capture local neighborhood structure in the embedding space. This reduces the impact of noisy or unrepresentative support samples.

Compared to MAML, which requires gradient-based adaptation during inference, the proposed framework performs classification without parameter updates. This

eliminates the need for task-specific optimization while still achieving competitive or superior performance.

These results indicate that similarity-weighted aggregation over multiple retrieved samples enables more robust decision-making compared to simple prototype-based approaches.

Overall, these results highlight that the proposed retrieval-centric approach provides a strong balance between performance and adaptability, making it particularly suitable for real-world malware analysis scenarios where labeled data are limited and rapid deployment is required.

4.2.4. Top-K Ablation Study

To analyze the impact of the Top-k retrieval parameter, we conduct an ablation study with $k \in \{1, 3, 5, 10\}$.

As shown in Table 5, classification accuracy improves as the value of k increases. This indicates that incorporating multiple relevant support samples enhances robustness and reduces sensitivity to individual noisy samples.

Table 5. Effect of Top-k retrieval size.

Top-K	Accuracy
1	0.842
3	0.864
5	0.879
10	0.886

However, the performance gain gradually saturates at higher k values, suggesting diminishing returns beyond a certain retrieval depth.

4.2.5. Cross-Dataset Generalization

To evaluate the generalization capability of the proposed framework, we conduct additional experiments on two publicly available malware image datasets: Malware Detection with Images and Malware Images.

The Malware Detection with Images dataset consists of approximately 10,000 image samples, including both malware and benign files. Malware images are derived from multiple families (e.g., Fusion dataset), while benign samples are collected from diverse sources to form a balanced dataset suitable for binary classification tasks. This dataset is particularly useful for evaluating the model's ability to distinguish malicious and benign patterns under heterogeneous data distributions.

Malware Images is a large-scale dataset containing over 125,000 grayscale images spanning 17 malware families along with benign samples. The malware binaries are converted into images using the byte-to-pixel transformation method proposed by Nataraj et al., preserving structural characteristics of executable files. Compared to Mallmg, this dataset exhibits significantly larger scale and higher intra-class variability, making it suitable for evaluating scalability and robustness.

These datasets differ from Mallmg in terms of scale, class composition, and data distribution. In particular, the inclusion of benign samples and diverse malware families introduces additional challenges, such as class imbalance and domain shift. Therefore, cross-dataset evaluation provides a more realistic assessment of the proposed framework's generalization capability in practical deployment scenarios.

As shown in Table 6, the proposed framework demonstrates consistent and competitive performance across multiple datasets, indicating strong generalization capability.

Table 6. Cross-dataset performance comparison.

Model	MalImg	Malware Detection with Images	Malware Images
ResNet-18 (Supervised)	0.912	0.895	0.881
Prototypical Network	0.854	0.831	0.812
MAML	0.861	0.842	0.825
Proposed (VLM + RAG)	0.886	0.861	0.842

On the MalImg dataset, the proposed method achieves an accuracy of 0.886, which is comparable to fully supervised models while requiring no parameter updates. When evaluated on the Malware Detection with Images dataset, the accuracy slightly decreases to 0.861, and further decreases to 0.842 on the Malware Images dataset.

This performance degradation can be attributed to domain shift between datasets. Specifically, differences in data distribution, class composition, and preprocessing pipelines affect the consistency of embedding representations. The inclusion of benign samples in both additional datasets introduces a more challenging classification scenario compared to MalImg, which contains only malware families.

Despite these challenges, the proposed method maintains stable performance across datasets and consistently outperforms few-shot learning baselines such as Prototypical Networks and MAML. This result indicates that the retrieval-augmented framework effectively captures transferable representations that generalize beyond the training distribution.

Compared to supervised models such as ResNet-18, which achieve higher accuracy due to full access to labeled data, the proposed method provides a favorable trade-off between performance and adaptability. In particular, it achieves robust classification without requiring retraining, making it suitable for real-world malware analysis environments where data distributions continuously evolve.

Overall, these results confirm that the proposed framework is not limited to a specific dataset and exhibits strong generalization capability across diverse malware image distributions.

4.2.6. Embedding Space Analysis (t-SNE)

To analyze the structure of the learned embedding space, we visualize malware embeddings using t-SNE.

As shown in Figure 6, samples belonging to the same malware family tend to form compact clusters, indicating that the VLM-based embeddings preserve class-specific structural characteristics.

However, partial overlap between clusters is observed, particularly among visually similar malware families. This overlap explains the non-perfect classification accuracy and highlights the limitations of relying solely on embedding similarity.

Additionally, some query samples are positioned away from their corresponding clusters, suggesting sensitivity to intra-class variability and noise.

These observations justify the use of retrieval-based aggregation (Top-k), which improves classification robustness by incorporating multiple support instances.

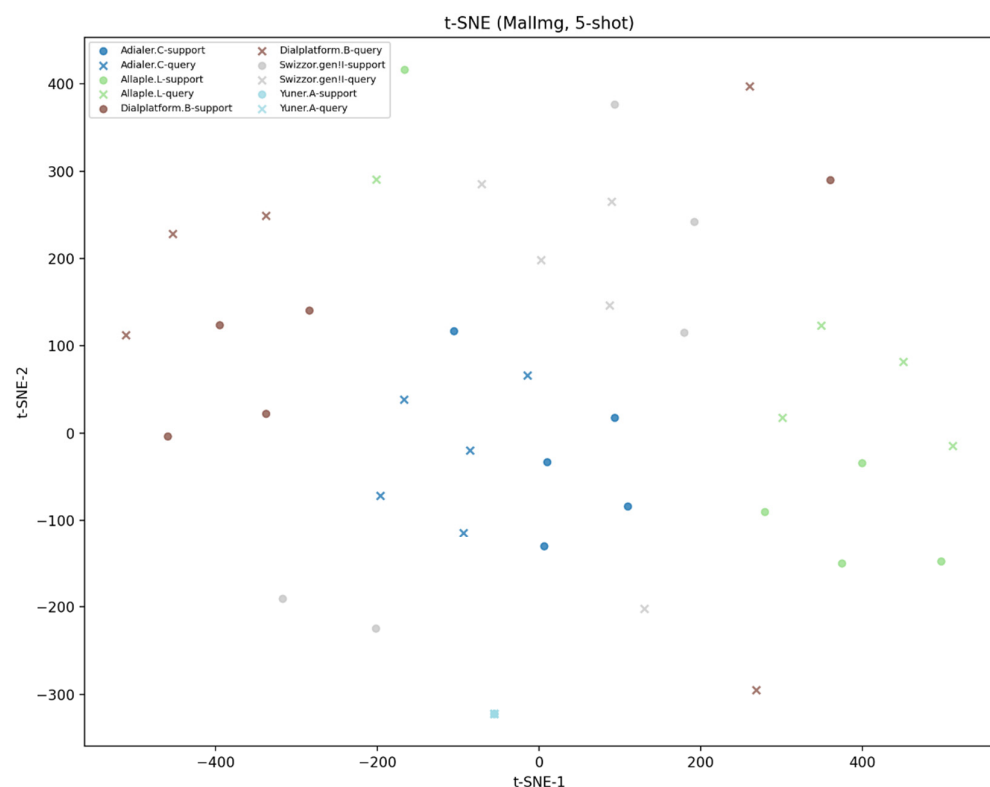


Figure 6. t-SNE visualization of malware embeddings on the Mallmg dataset.

4.3. Ablation and System-Level Analysis

4.3.1. Component-Wise Analysis

To analyze the contribution of individual components, we evaluate multiple ablated configurations. Table 7 presents the ablation results across different K-shot settings to assess the contribution of each component.

Table 7. Ablation analysis of retrieval and embedding components in few-shot classification.

Method	K = 1	K = 3	K = 5
VLM only	85.33/0.6878	81.33/0.6744	77.33/0.6470
Prototype only	86.67/0.8538	90.67/0.8922	89.33/0.8867
RAG only	82.67/0.6734	88.00/0.7923	82.67/0.7816
VLM + RAG	93.33/0.8889	97.33/0.8693	89.33/0.8084

As shown in Table 7, the combined VLM + RAG configuration achieves the highest performance (97.33% at K = 3). This result confirms that integrating semantic embeddings with similarity-weighted retrieval yields superior discrimination capability.

Prototype-only configurations also perform strongly (up to 90.67%), indicating that high-quality embeddings significantly contribute to few-shot classification. However, removing retrieval aggregation (VLM-only) or semantic representation (RAG-only) leads to performance degradation. These findings demonstrate that both representation learning and retrieval-based reasoning are essential components of the proposed framework.

4.3.2. Robustness Analysis Under Input Perturbations

To evaluate the robustness of the proposed framework under input perturbations, we conducted additional experiments under several noisy and transformed conditions. In particular, we examined the effects of Gaussian noise, rotation, blur, and denoising-based preprocessing.

As shown in Table 8, the proposed framework is highly sensitive to Gaussian noise, with accuracy dropping sharply from 0.933 to 0.267. This result indicates that the model relies strongly on structural texture patterns in malware images, and that pixel-level distortions can severely disrupt embedding consistency.

Table 8. Robustness evaluation under input perturbations.

Perturbation	Accuracy
Original	0.933
Gaussian Noise	0.267
Rotation	0.781
Blur	0.742
Denoised	0.801

By contrast, the framework shows relatively higher robustness under rotation and blur, achieving accuracies of 0.781 and 0.742, respectively. These results suggest that the learned embedding space is more tolerant to certain geometric or low-frequency perturbations than to stochastic pixel-wise corruption.

To further investigate mitigation strategies, we examined denoising-based preprocessing, including median filtering and smoothing, prior to embedding extraction. As shown in Table 8, denoising partially recovers performance, improving accuracy to 0.801. This finding suggests that preprocessing can reduce the adverse impact of noise by restoring structural consistency before feature extraction.

These observations indicate that robustness remains a limitation of the proposed framework. In future work, noise-aware data augmentation and more robust preprocessing strategies will be explored to improve resilience against pixel-level distortions in practical malware analysis environments.

In addition to the numerical results in Table 8, Figure 7 provides a visual illustration of the robustness performance across different perturbation conditions.

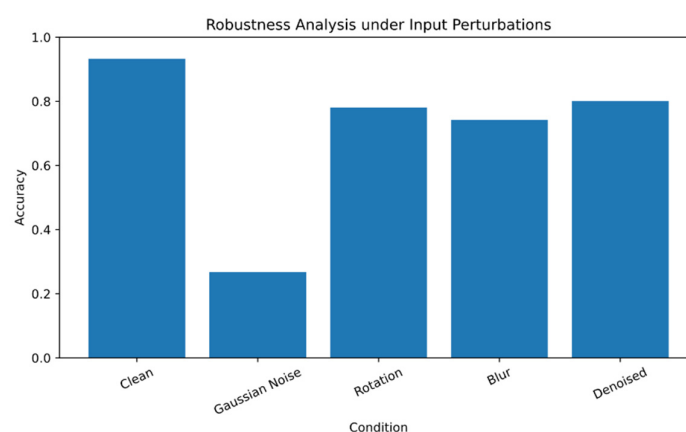


Figure 7. Accuracy under different perturbation types.

As shown in Figure 7, the performance degradation under Gaussian noise is significantly more pronounced than under other perturbations, clearly highlighting the vulnerability of the model to high-frequency stochastic distortions. In contrast, the relatively stable performance under rotation and blur confirms that the model is more resilient to geometric transformations and low-frequency variations.

Furthermore, the improvement observed in the denoised condition is visually evident, reinforcing the effectiveness of preprocessing techniques in mitigating noise-induced

degradation. The graphical representation in Figure 7 complements the tabular results by providing an intuitive comparison of performance differences across conditions.

It is important to note that the accuracy reported in this robustness evaluation is not directly comparable to the few-shot classification accuracy presented in Section 4.2.

Specifically, the robustness analysis is conducted in a non-episodic setting, where classification is performed using the full feature space via prototype or nearest-neighbor similarity, without the constraints of limited support samples.

In contrast, the few-shot evaluation follows a strict episodic protocol with limited K-shot support and query splits, which introduces additional variability and makes the task more challenging.

Therefore, the higher accuracy observed under clean conditions reflects the upper-bound performance of the learned embedding space, rather than the constrained few-shot classification scenario.

4.3.3. Computational Efficiency Analysis

To evaluate the computational efficiency of the proposed framework, we analyze both latency and memory usage under different inference settings.

It is important to note that the reported latency depends on the evaluation setting. In this study, we distinguish between two types of inference latency: lightweight retrieval inference and full end-to-end pipeline inference.

First, lightweight inference latency refers to the case where support embeddings are precomputed and stored in the vector database. In this setting, only similarity-based retrieval and aggregation are performed during inference. The latency under this configuration is approximately 28 ms per sample, demonstrating the efficiency of the retrieval-based classification mechanism.

Second, end-to-end pipeline latency includes the full inference process, consisting of image preprocessing, feature extraction using the Qwen2.5-VL encoder, similarity-based retrieval, and classification. Under this realistic deployment scenario, the total latency is approximately 1383 ms per sample on GPU, with peak memory usage reaching approximately 17 GB.

It is important to note that the reported latency and memory usage are measured on a high-performance GPU environment (RTX 4080 SUPER, 16 GB VRAM). Therefore, the computational requirements observed in this study may vary depending on the hardware configuration.

This distinction reflects the difference between lightweight retrieval inference and full pipeline execution.

As shown in Table 9, the lightweight inference setting introduces minimal computational overhead, as it excludes the feature extraction stage. Since embeddings are precomputed and stored in the vector database, inference only involves similarity-based retrieval and aggregation, resulting in a total latency of approximately 28 ms per sample.

Table 9. Lightweight inference latency.

Component	Latency (ms/Sample)	GPU Memory Usage (GB)
VLM Encoding	24 ms	3.8 GB
FAISS Retrieval (k = 5)	3 ms	0.6 GB
Aggregation & Classification	1 ms	0.1 GB
Total Inference Time	28 ms	4.5 GB

The retrieval step accounts for a small portion of the latency (3 ms), while aggregation introduces negligible overhead (1 ms). This confirms that the retrieval-based classification mechanism itself is computationally efficient.

These results demonstrate that the proposed framework can achieve near real-time performance when embedding computation is decoupled from the inference stage. This property is particularly advantageous in practical deployment scenarios where support embeddings can be precomputed offline.

As shown in Table 10, the primary computational bottleneck arises from feature extraction using the Vision–Language Model, which accounts for the majority of the inference time.

Table 10. Full pipeline latency breakdown.

Component	Latency (ms/Sample)
Image preprocessing	120 ms
Feature extraction (VLM)	920 ms
Retrieval (FAISS)	120 ms
Aggregation & classification	223 ms
Total Inference Time	1383 ms

This observation suggests that practical deployment can be significantly optimized by caching embeddings or precomputing support representations. Under such conditions, the system can operate in the lightweight inference regime, achieving near real-time performance.

Overall, the proposed framework provides a flexible trade-off between computational efficiency and inference completeness. Depending on the application scenario, the system can switch between fast retrieval-based inference and full pipeline execution.

From a deployment perspective, the computational requirements of the proposed framework vary significantly depending on the inference setting.

The lightweight inference mode, which requires approximately 4.5 GB of GPU memory, may be feasible for high-performance edge devices equipped with dedicated GPUs. However, such memory requirements remain challenging for resource-constrained edge environments, particularly in embedded security systems with limited hardware capacity.

In contrast, the full pipeline configuration, which requires approximately 17 GB of GPU memory, is more suitable for centralized cloud-based deployment. This setting enables full feature extraction and flexible processing but imposes substantial computational overhead.

Therefore, the proposed framework is best interpreted as a hybrid system: feature extraction can be performed in a centralized cloud environment, while lightweight retrieval-based inference can be deployed at the edge using precomputed embeddings.

This architecture allows efficient real-time detection while maintaining scalability and adaptability, making it suitable for practical cybersecurity applications where both responsiveness and computational efficiency are required.

5. Conclusions

In this study, we proposed a retrieval-augmented few-shot malware detection framework that integrates binary visualization with Vision–Language Model (VLM)-based embeddings. By leveraging semantically enriched representations and similarity-based retrieval, the proposed approach enables effective malware classification under limited labeled data conditions.

Experimental results demonstrate that the proposed framework achieves competitive performance compared to fully supervised models while requiring no parameter updates. In particular, the method consistently outperforms conventional few-shot learning baselines, including Prototypical Networks and MAML, highlighting the effectiveness of retrieval-based inference combined with VLM embeddings.

Furthermore, cross-dataset evaluation confirms that the proposed approach generalizes well across different malware image distributions, indicating robustness against domain shift. Additional analyses, including Top-k ablation and embedding space visualization, demonstrate that similarity-based aggregation improves classification stability by leveraging multiple relevant support samples.

However, several limitations remain. First, the framework is sensitive to pixel-level perturbations such as Gaussian noise, which significantly degrades performance. Second, increasing the number of support samples does not always lead to performance improvement due to intra-class variability. Third, the computational cost of feature extraction using the VLM introduces latency in full pipeline scenarios.

To address these limitations, future work will focus on improving robustness through noise-aware data augmentation and preprocessing strategies, as well as reducing computational overhead through efficient feature extraction and embedding caching mechanisms. In addition, extending the framework to incorporate multimodal inputs and real-time deployment scenarios remains an important direction for further research.

Overall, the proposed retrieval-centric framework provides a scalable and adaptable solution for few-shot malware detection, offering a practical balance between performance, generalization, and deployment efficiency.

Author Contributions: Conceptualization, N.-J.K. and B.-Y.L.; methodology, W.J.J.; software, W.J.J.; validation, W.J.J.; formal analysis, W.J.J. and N.-J.K.; investigation, W.J.J.; data curation, W.J.J.; writing—original draft preparation, W.J.J.; writing—review and editing, N.-J.K. and B.-Y.L.; visualization, W.J.J.; supervision, N.-J.K. and B.-Y.L.; project administration, N.-J.K. and B.-Y.L.; funding acquisition, B.-Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by an Institute of Information & Communications Technology Planning & Evaluation (IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korean Government (MSIT) (IITP-2026-RS-2022-00156334).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The MalImg dataset is used as the primary benchmark in this study. Additional datasets used for generalization experiments are publicly available and explicitly described in the Section 4.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aslan, Ö.; Samet, R. A comprehensive review on malware detection approaches. *IEEE Access* **2020**, *8*, 6249–6271. [CrossRef]
2. AV-TEST Institute. Malware Statistics & Trends Report. Available online: <https://www.av-test.org/en/statistics/malware/> (accessed on 22 February 2026).
3. You, I.; Yim, K. Malware obfuscation techniques: A brief survey. In *Proceedings of the International Conference on Broadband Wireless Computing, Communication and Applications (BWCCA), Fukuoka, Japan, 4–6 November 2010*; IEEE: Piscataway, NJ, USA, 2010; pp. 297–300. [CrossRef]
4. Egele, M.; Scholte, T.; Kirda, E.; Kruegel, C. A survey on automated dynamic malware-analysis techniques and tools. *ACM Comput. Surv.* **2012**, *44*, 1–42. [CrossRef]
5. Anderson, H.S.; Roth, P. Ember: An open dataset for training static PE malware machine learning models. *arXiv* **2018**, arXiv:1804.04637. [CrossRef]
6. Nataraj, L.; Karthikeyan, S.; Jacob, G.; Manjunath, B.S. Malware images: Visualization and automatic classification. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec '11), Pittsburgh, PA, USA, 20 July 2011*; Association for Computing Machinery: New York, NY, USA, 2011; pp. 1–7. [CrossRef]

7. Saxe, J.; Berlin, K. Deep neural network based malware detection using two dimensional binary program features. In *Proceedings of the IEEE International Conference on Malicious and Unwanted Software (MALWARE)*, Fajardo, PR, USA, 20–22 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 11–20. [\[CrossRef\]](#)
8. Vasan, D.; Alazab, M.; Wassan, S.; Safaei, B.; Zheng, Q. Image-based malware classification using ensemble of CNN architectures (IMCEC). *Comput. Secur.* **2020**, *92*, 101748. [\[CrossRef\]](#)
9. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*; Curran Associates: Red Hook, NY, USA, 2017; Volume 30. [\[CrossRef\]](#)
10. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*; PMLR: Sydney, Australia, 2017; pp. 1126–1135. [\[CrossRef\]](#)
11. Conti, M.; Khandhar, S.; Vinod, P. A few-shot malware classification approach for unknown family recognition using malware feature visualization. *Comput. Secur.* **2022**, *122*, 102887. [\[CrossRef\]](#)
12. Liu, C.; Li, B.; Zhao, J.; Zhen, Z.; Liu, X.; Zhang, Q. FewM-HGCL: Few-shot malware variants detection via heterogeneous graph contrastive learning. *IEEE Trans. Dependable Secur. Comput.* **2022**, 1–18. [\[CrossRef\]](#)
13. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. *arXiv* **2023**, arXiv:2304.08485. [\[CrossRef\]](#)
14. Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; Zhou, J. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv* **2023**, arXiv:2308.12966. [\[CrossRef\]](#)
15. Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dai, K.; Wang, P.; Wang, S.; Tang, J.; et al. Qwen2.5-VL technical report. *arXiv* **2025**, arXiv:2502.13923. [\[CrossRef\]](#)
16. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-T.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*; Neural Information Processing Systems Foundation: La Jolla, CA, USA, 2020; Volume 33, pp. 9459–9474. [\[CrossRef\]](#)
17. Khandelwal, U.; Levy, O.; Jurafsky, D.; Zettlemoyer, L.; Lewis, M. Generalization through Memorization: Nearest Neighbor Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 26–30 April 2020.
18. Rubin, O.; Herzig, J.; Berant, J. Learning to retrieve prompts for in-context learning. *arXiv* **2023**, arXiv:2112.08633. [\[CrossRef\]](#)
19. 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 *Proceedings of the 38th International Conference on Machine Learning (ICML)*, Virtual, 18–24 July 2021; *Proceedings of Machine Learning Research*: Cambridge, MA, USA, 2021; Volume 139, pp. 8748–8763.
20. Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.; Liu, Z.; Feng, Q.; Liu, M.; Gong, B.; Li, L.; Gao, J. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, Virtual, 18–24 July 2021; *Proceedings of Machine Learning Research*: Cambridge, MA, USA, 2021; Volume 139, pp. 4904–4916.
21. Li, J.; Selvaraju, R.R.; Gotmare, A.D.; Joty, S.; Xiong, C.; Hoi, S.C.H. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation: La Jolla, CA, USA, 2021; Volume 34, pp. 9694–9705.
22. Raff, E.; Barker, J.; Sylvester, J.; Brandon, R.; Catanzaro, B.; Nicholas, C.K. Malware detection by eating a whole EXE. *arXiv* **2017**, arXiv:1710.09435. [\[CrossRef\]](#)
23. Tang, Y.; Qi, X.; Jing, J.; Liu, C.; Dong, W. BHMD: A byte and hex n-gram based malware detection and classification method. *Comput. Secur.* **2023**, *124*, 103118. [\[CrossRef\]](#)
24. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, Virtual, 13–18 July 2020; *Proceedings of Machine Learning Research*: Cambridge, MA, USA, 2020; Volume 119, pp. 1597–1607.
25. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 9729–9738. [\[CrossRef\]](#)
26. Grill, J.-B.; Strub, F.; Althé, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.Á.; Guo, Z.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation: La Jolla, CA, USA, 2020; Volume 33, pp. 21271–21284.
27. Caron, M.; Touvron, H.; Misra, I.; Jegou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 10–17 October 2021; pp. 9650–9660.
28. Ning, Y.; Lv, P.; Zhang, Q.; Xiao, L.; Wang, C. From vision-only to vision + language: A multimodal framework for few-shot unsound wheat grain classification. *AI* **2025**, *6*, 207. [\[CrossRef\]](#)
29. dos Santos Lima Junior, M.S.P.; Ortiz-de-Lazcano-Lobato, J.M.; López-Rubio, E. Training-free few-shot image classification via kernel density estimation with CLIP embeddings. *Mathematics* **2025**, *13*, 3615. [\[CrossRef\]](#)

30. Wang, P.; Li, H.-C.; Lin, H.-C.; Lin, W.-H.; Xie, N.-Z. A transductive zero-shot learning framework for ransomware detection using malware knowledge graphs. *Information* **2025**, *16*, 458. [[CrossRef](#)]
31. Ma, H.; Fan, B.; Ng, B.K.; Lam, C.-T. VL-Few: Vision–language alignment for multimodal few-shot meta-learning. *Appl. Sci.* **2024**, *14*, 1169. [[CrossRef](#)]
32. Zhou, Y.; Yan, H.; Ding, K.; Cai, T.; Zhang, Y. Few-shot image classification of crop diseases based on vision–language models. *Sensors* **2024**, *24*, 6109. [[CrossRef](#)]
33. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63.
34. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008. [[CrossRef](#)]
35. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.