Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.068201

ARTICLE

Proactive Disentangled Modeling of Trigger–Object Pairings for Backdoor Defense

Kyle Stein[1,*], **Andrew A. Mahyari**[1,2], **Guillermo Francia III**[3] **and Eman El-Sheikh**[3]

[1]Department of Intelligent Systems and Robotics, University of West Florida, Pensacola, FL 32514, USA
[2]Florida Institute for Human and Machine Cognition (IHMC), Pensacola, FL 32502, USA
[3]Center for Cybersecurity, University of West Florida, Pensacola, FL 32502, USA

*Corresponding Author: Kyle Stein. Email: ks209@students.uwf.edu

Received: 23 May 2025; Accepted: 11 July 2025; Published: 29 August 2025

ABSTRACT: Deep neural networks (DNNs) and generative AI (GenAI) are increasingly vulnerable to backdoor attacks, where adversaries embed triggers into inputs to cause models to misclassify or misinterpret target labels. Beyond traditional single-trigger scenarios, attackers may inject multiple triggers across various object classes, forming unseen backdoor-object configurations that evade standard detection pipelines. In this paper, we introduce **DBOM** (**D**isentangled **B**ackdoor-**O**bject **M**odeling), a proactive framework that leverages structured disentanglement to identify and neutralize both seen and unseen backdoor threats at the dataset level. Specifically, DBOM factorizes input image representations by modeling triggers and objects as independent primitives in the embedding space through the use of Vision-Language Models (VLMs). By leveraging the frozen, pre-trained encoders of VLMs, our approach decomposes the latent representations into distinct components through a learnable visual prompt repository and prompt prefix tuning, ensuring that the relationships between triggers and objects are explicitly captured. To separate trigger and object representations in the visual prompt repository, we introduce the trigger–object separation and diversity losses that aids in disentangling trigger and object visual features. Next, by aligning image features with feature decomposition and fusion, as well as learned contextual prompt tokens in a shared multimodal space, DBOM enables zero-shot generalization to novel trigger-object pairings that were unseen during training, thereby offering deeper insights into adversarial attack patterns. Experimental results on CIFAR-10 and GTSRB demonstrate that DBOM robustly detects poisoned images prior to downstream training, significantly enhancing the security of DNN training pipelines.

KEYWORDS: Backdoor attacks; generative AI; disentanglement

1 Introduction

As deep neural networks (DNNs) become more prevalent in applications such as natural language processing [1–3] and object classification [4–6], they are increasingly being targeted by sophisticated security threats [7,8]. The rise of generative AI [9–11] has enabled the large-scale creation of datasets sourced from online repositories. Although these datasets improve model robustness, they often bypass rigorous vetting, making them vulnerable to backdoor attacks [12–15]. Such attacks embed hidden triggers in training samples, causing models to misclassify inputs containing the trigger, for example, altering a stop sign's label to a speed limit sign.

Recent work has focused on identifying backdoored samples in pre-trained infected models [16–19], but less attention has been given to proactively scanning training data for suspicious triggers before the final model is trained. This lack of focus on the dataset creation phase represents a significant gap in input-level backdoor defense strategies [20–23]. Malicious triggers can be embedded in training samples well before the model is exposed to them, undermining the integrity of the entire training process. Addressing this stage early in the pipeline not only prevents contaminated data from infiltrating the training process, but also reduces the computational costs associated with post-training purification efforts [24,25]. Lastly, proactively analyzing the dataset offers deeper insights into the adversarial logic behind these backdoors, specifically how triggers interact with objects and how attackers strategically embed them to exploit vulnerabilities.

Although existing defenses can detect single or multiple backdoor triggers in a compromised data set [26–30], they remain strictly trigger-centric, where flagged samples are discarded, and images of objects classes bearing those triggers are ignored. This removes valuable co-occurrence information into how specific triggers map onto particular objects, which could expose systematic attacker strategies. In realistic many-to-many attack scenarios [31], where adversaries plant various triggers across a wide range of object categories, a trigger-only approach would fail to recognize novel trigger-object combinations outside of its training set of known trigger–object pairings. For instance, assume a square-patch trigger is only ever seen on stop signs and a pixel-noise trigger only on speed-limit signs. If an attacker then applies that same square patch to yield signs or the pixel noise to pedestrian-crossing signs (pairings never observed before) those trigger-centric detectors may sharply degrade in performance, since they do not explicitly model which object the trigger appears on. By contrast, a co-occurrence-aware model that simultaneously identifies both triggers and object classes preserves the relational context between adversarial triggers and their targets. Rather than excluding compromised samples, this approach leverages modular relationships to learn comprehensive backdoor patterns and infer previously unseen trigger–object combinations. As a result, the model can accurately recognize the underlying object despite the presence of a trigger, integrate attacked examples into both training and inference workflows, and reduce false positives by distinguishing benign from malicious features. Moreover, modeling trigger–object relationships provides deeper forensic insights into attacker tactics, enabling dynamic update strategies that proactively defends models against evolving many-to-many backdoor attacks. Overall, we can summarize that existing input-level defenses in current state-of-the-art (SOA) attack scenarios remain strictly trigger-centric, where: (1) they identify and discard adversarial samples, losing the underlying object semantics and missing the opportunity to reveal adversarial strategies, (2) do not focus on concurrently identifying triggers and the associated object class, and (3) fail to generalize to novel trigger-object pairings.

To address these gaps, we present Disentangled Backdoor-Object Modeling (**DBOM**), a proactive framework based on VLMs and prompt tuning [9], designed to identify and isolate unseen backdoor-object configurations. Instead of inspecting a potentially compromised model, this approach focuses on learning trigger-object configurations within web-scraped training images before they are ever fed into a downstream model. Our method surpasses current SOA pre-training defense algorithms by detecting not only the types of backdoor triggers in compromised datasets, but also the underlying objects they target, thereby capturing the adversarial logic behind these malicious trigger–object pairings. Here, we define a trigger as the backdoor attack pattern embedded into an image and an object as the benign semantic class being manipulated. DBOM then factorizes these two primitives into independent embeddings (Fig. 1), enabling modular representations of trigger–object configurations [32]. Furthermore, by capturing the relationship among triggers and objects during training, **previously unseen trigger-object pairings can be detected during inference**, a problem traditional single-trigger detection pipelines overlook. The contributions of our approach are as follows:

- We introduce DBOM, a novel end-to-end disentangled representation learning framework that separates triggers and objects into independent latent visual primitives. By leveraging cross-modal attention for structured latent decomposition, DBOM aims to learn each trigger pattern and each object class in isolation. At inference, it recomposes these known trigger and object embeddings to recognize combinations never seen during training, achieving zero-shot generalization over trigger–object pairings and resulting in a robust method against adaptive backdoor strategies.
- Our approach incorporates a dual-branch module that features a learnable visual prompt repository along with a dynamic soft prompt prefix adapter for prompt tuning. The use of a learnable visual prompt repository allows us to capture primitive-specific features for both triggers and objects, aiding in feature disentanglement. Furthermore, dynamically tuning text prompt representations based on image content, our module enhances the semantic context of each sample and improves the separation between trigger and object features. This design allows the framework to capture diverse trigger patterns across multiple object classes, overcoming the limitations of conventional defenses that assume a single, static trigger per dataset.
- By integrating a proactive backdoor detection mechanism into the data curation process, DBOM identifies unseen backdoor-object attacks before downstream model training begins. A composite loss function that minimizes cross-entropy, disentanglement, and prompt alignment losses together ensures that poisoned samples are identified and isolated for removal from the dataset.

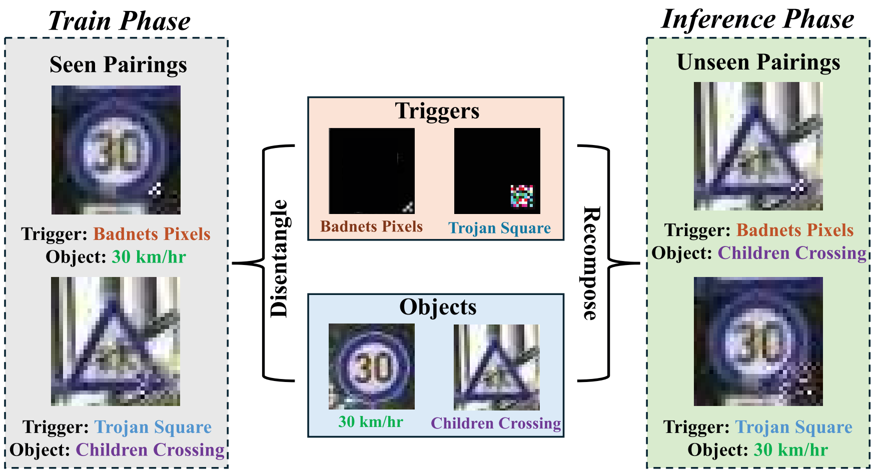

Figure 1: Overview of our disentangling process for trigger–object pairings. During training, the system learns separate representations of triggers and objects. By factorizing these components, the model can generalize to unseen trigger–object configurations, although they were never observed together during training

2 Related Work

Disentanglement involves separating visual primitives of images into independent components [33–37]. A central strategy for addressing this task is to train models that learn these independent components and recombine them in novel ways, thereby enabling the flexible recognition of previously unseen trigger–object pairings. Li et al. [14] apply symmetry and group theory to model primitive relationships, introducing a novel distance function. A Siamese Contrastive Embedding Network (SCEN) [38] embeds visual features into a contrastive space to separately model primitive diversity. A retrieval-augmented approach improves recognition of unseen primitive component pairings by retrieving and refining representations [39]. Recent methods integrate vision-language models (VLMs) such as CLIP [9] to enhance the recognition of structured relationships between the underlying nature of images and text prompts. Compositional Soft Prompting (CSP) [40] utilizes a static prompt prefix alongside learned primitive embeddings, with predictions based on cosine similarity between text and image features. Later works remove the static prefix, making the entire prompt learnable [41,42]. In the context of DBOM, disentangling triggers and objects allows our model to factor visual embeddings into two primitive subspaces: one that captures adversarial trigger patterns and one that encodes the class object semantics. Once these primitives are learned, unseen trigger-object pairings can be inferred upon during testing.

Backdoor Attacks became prominent with the introduction of Badnets [12]. Badnets demonstrated how adversaries can embed backdoors into DNNs by poisoning the training data with trigger-patterned images, such as a single white square or pixelated patterns, to misclassify inputs. Liu et al. [13] introduced trojaning attacks, which differ from Badnets, by reverse-engineering neuron activations to generate adversarial triggers that maximize activations in specific neurons. Li et al. [43] explored techniques to make triggers more covert to detection by implementing steganographic embedding, where backdoor triggers are hidden within images at a pixel level. Recent backdoor attacks include Wanet [15], a warping-based trigger, which introduces imperceptible image distortions as triggers instead of traditional noise perturbations.

Backdoor Defenses mostly operate in the adversarial machine learning life-cycle at the model level, leaving the dataset vetting process largely unexplored [44]. Several works attempt to filter adversarial images before training [20–22,29], but these rely on detecting known trigger-object configurations and fail to generalize to unseen pairings. VisionGuard [21] compares the softmax outputs of original and transformed images using metrics like the Kullback–Leibler divergence to detect attacks without altering the target network. Deep k-NN [20] leverages deep feature space clustering and k-nearest neighbor voting to detect and remove poisoned images from the training set prior to downstream model training. HOLMES [22] employs multiple external detectors trained on both dedicated labels and top-k logits to capture subtle differences between benign and adversarial inputs. Traditional backdoor defenses assume a compromised model and attempt to mitigate attacks post-training [17–19]. However, these techniques reactively address attacks after deployment by cleaning the model, whereas our approach proactively filters poisoned images before they enter the downstream training pipeline, preventing backdoor contamination at its source. Furthermore, these methods overlook the opportunity to identify unseen trigger–object configurations that were not seen in their model training, which is addressed in this paper.

3 Preliminaries and Insights

3.1 Trigger-Object Representation

We define a backdoor configuration as a pairing of a *trigger* and an *object*, where the trigger serves as the adversarial modification and the object represents the underlying semantic class being targeted (e.g., "stop sign," "yield sign," "airplane"). Let T be the set of all possible triggers, and O be the set of object categories, where $T = \{t_0, t_1, \ldots, t_n\}$ and $O = \{o_0, o_1, \ldots, o_m\}$. The complete set of potential trigger–object pairings is given by $P = T \times O$, where each pair $(t, o) \in P$ corresponds to a unique backdoor attack configuration. These pairings can be categorized into two groups: (1) *seen pairings* (P_s), which are explicitly observed during training, and (2) *unseen pairings* (P_u), which do not appear in the training set but may still be encountered during deployment. These subsets are disjoint ($P_s \cap P_u = \varnothing$) and together form the complete space of possible attack configurations ($P_s \cup P_u = P$). During evaluation, test samples are drawn from a predefined set $P_{\text{test}} \subseteq P$, which contains both seen and unseen pairings. The objective of our approach is to learn a function $f : X \rightarrow P_{\text{test}}$, where X represents the input space of images containing these trigger–object configurations. The function f is designed to map an image to its corresponding attack configuration, enabling generalization to **unseen trigger–object pairs** that were not part of the training distribution. Furthermore, we note that the goal of this paper is not to train an infected model or defend against attacked models, but to detect backdoored images before downstream model training begins.

3.2 Threat Model and Defender Goals

Threat Model. We assume an adversary injects backdoor attacks based on trigger–object pairings into a web-scraped or publicly available dataset used for training a downstream DNN. The goal is to cause the model to misclassify inputs containing triggers into a target label while maintaining normal classification on clean images. Since large datasets are rarely vetted on a per-sample basis, malicious samples blend easily with clean data. Furthermore, attackers can escalate this threat by injecting multiple triggers across different classes, including novel, unseen trigger–object pairings, so that conventional defenses which expect a single static trigger fail to detect them. Consequently, the compromised data is used in downstream training, embedding hidden adversarial behaviors into the final model.

Defender's Goal. The defender's goal is to *identify backdoored images prior to downstream model training*, ensuring they are isolated while minimizing the misclassification of clean images. Given a potentially poisoned dataset that contains several triggers–object configurations, the defender must distinguish legitimate images from those carrying triggers. Furthermore, by concurrently identifying both the trigger and the underlying object, the defender learns vital information into the adversary's strategies. Moreover, separating the adversarial trigger from the underlying object enables the recovery of correct object semantics in backdoored samples, eliminating the need to discard these adversarial samples from training or inference.

4 Proposed Framework

DBOM leverages CLIP as its backbone by freezing its pre-trained visual and text encoders. Let $f_\theta(\cdot)$ denote the CLIP image encoder and $g_\phi(\cdot)$ denote the CLIP text encoder. Given an input image x_i, the image encoder extracts visual features $f_v = f_\theta(x_i) \in \mathbb{R}^d$, which serve two purposes: (i) they are used to retrieve the most relevant visual prompts from a learnable repository, and (ii) they provide the bias for shifting a set of learnable prefix text tokens `[v1] [v2] [v3]` via a prompt adapter network. Unlike fixed prefix templates (i.e., `a photo of`), our approach employs *prompt tuning*, a technique where these prefix tokens are treated as learnable parameters and optimized end-to-end to capture task-specific context for each image. This allows the text prompt to be tailored to the visual content of each image, promoting the alignment between visual and textual modalities. The shifted prefix is then appended to the trigger and object word embeddings to form the final prompt t_i, which is processed by the text encoder to produce text features $f_t = g_\phi(t_i) \in \mathbb{R}^{768}$. Lastly, f_v and f_t are decomposed and fused, and their joint representation is mapped into a separate pair space where the similarity between the image and fused features helps determine the final trigger-object prediction. Fig. 2 displays the overall architecture of the proposed approach.

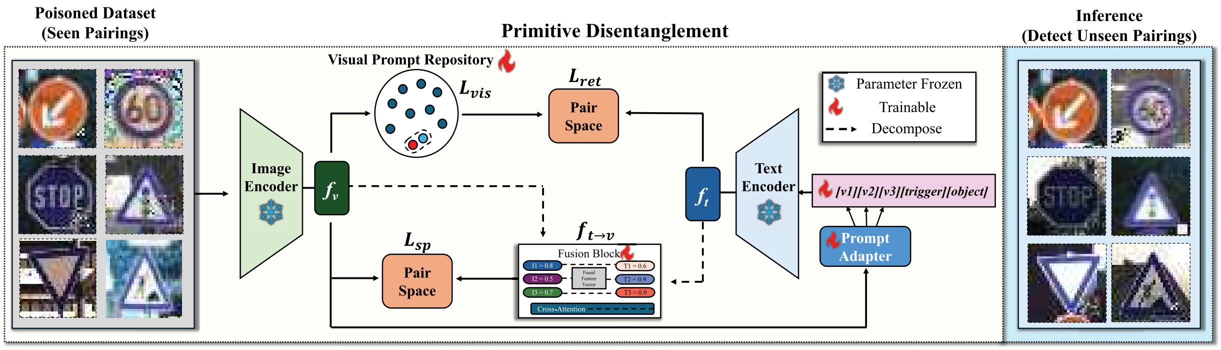

Figure 2: DBOM utilizes a visual prompt repository and a similarity-based retrieval mechanism to detect unseen backdoor trigger-object representations through the use of CLIP's pre-trained visual and textual encoders. During training, each image retrieves visual prompts from the repository, shifts a learnable text prefix with a prompt adapter, and fuses decomposed image–text features via cross-attention. During inference, the framework again retrieves the top visual prompts, shifts the text prompt for each new image, and computes similarity scores to pinpoint unseen trigger-object pairings. Lastly, in separate pair spaces, the logits are computed by comparing the fused image–text features with the visual features from the frozen visual encoder, as well as the selected visual prompts and the text features from the frozen text encoder. The highest-scoring trigger–object pair is then selected as the predicted configuration. By detecting malicious seen and unseen configurations in this way, DBOM identifies backdoored configurations and isolates them for removal prior to downstream model training

4.1 Visual Prompt Repository

The visual prompt repository comprises a collection of M learnable visual prompts $\{\mathbf{P}_i\}_{i=1}^{M}$, with each prompt $\mathbf{P}_i \in \mathbb{R}^{l\times d}$ paired with a learnable key $\mathbf{a}_i \in \mathbb{R}^{d}$. These prompts capture high-level visual semantics and are refined during training. For a given image, cosine similarity is computed between the normalized image features f_v and each normalized key. Based on the similarity scores, the two most similar prompts are selected. One is intended to align with the image's **trigger** and the other with the **object**. To enforce this specialization, we introduce two auxiliary losses. The *trigger-object separation loss* is formulated as:

$$\mathcal{L}_{\text{sep}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp\left(\cos(f_v^{(i)}, \mathbf{a}_{\text{trig}}^{(i)})\right)}{\exp\left(\cos(f_v^{(i)}, \mathbf{a}_{\text{trig}}^{(i)})\right) + \exp\left(\cos(f_v^{(i)}, \mathbf{a}_{\text{obj}}^{(i)})\right)}\right). \tag{1}$$

Because our primary objective is to accurately flag backdoored images, the loss function prioritizes the trigger key by encouraging it to achieve a higher similarity score than the object key, with the object serving as complementary context for the image. The *visual prompt diversity loss* is defined as:

$$\mathcal{L}_{\text{div}} = \frac{1}{N}\sum_{i=1}^{N}\max\left(0,\ m\ -\ \cos\left(\mathbf{a}_{\text{trig}}^{(i)}, \mathbf{a}_{\text{obj}}^{(i)}\right)\right), \tag{2}$$

where $m = 0.5$ is a fixed margin. This term penalizes any excessive similarity between the retrieved trigger and object visual prompts, thereby promoting disentangled features for more distinct representations [45]. Combining these terms yields:

$$\mathcal{L}_{\text{vis}} = \mathcal{L}_{\text{sep}} + \mathcal{L}_{\text{div}}, \tag{3}$$

which guides the prompts to distinctly capture trigger and object characteristics. During training, the visual prompt repository is updated end-to-end with $\mathcal{L}_{\text{vis}}$. This ensures that the repository vectors are not static but are continuously refined to distinguish between trigger and object features. The final representation of the retrieved visual prompts can be denoted by f_{ret}.

4.2 Dynamic Prefix Adapter

Traditional prompt tuning approaches [9,40,46] use a fixed soft prompt prefix, where a sequence such as `[trigger] [object]` is appended with an initialized phrase `a photo of`. This means that the same prefix is applied to every sample, regardless of the unique characteristics of the trigger or object in the image. This prefix rigidity can hinder the system's ability to accurately distinguish between different trigger–object pairs. Motivated by the work in [46], we propose an adaptive prompt network module that dynamically adjusts the learnable prefix tokens based on the visual content of the input image. This has been shown to transfer the frozen backbone's generalization power to entirely new tasks with very few labeled examples [46–48].

Specifically, the prompt adapter utilizes the image features f_v to compute a bias term that is added to the base prompt tokens, thus tailoring the prompt to each individual sample. Besides, by dynamically shifting the soft-prompt prefix based on each image's visual features, the prompt prefix adapter aligns the text embeddings more closely with the specific trigger and object primitives, which in turn lets the model accurately recombine those known primitives into novel, unseen pairings at inference, improving zero-shot pairing performance. The prompt adapter is implemented as a lightweight neural network defined by:

$$\text{APNet}(f_v) = \mathbf{W}_2 \cdot \sigma(\mathbf{W}_1 \cdot f_v + \mathbf{b}_1) + \mathbf{b}_2, \tag{4}$$

where $\sigma(\cdot)$ denotes the ReLU activation function, and $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are trainable parameters. The output, $\varphi(f_v)$, represents the bias added element-wise to the original prompt embeddings $\{\theta_0, \theta_1, \ldots, \theta_p\}$ via $\theta_i' = \theta_i + \varphi_i(f_v)$ for $i = 0, \ldots, p$. The final text prompt t_i is constructed by appending $\{\theta_0', \theta_1', \ldots, \theta_p'\}$ with the trigger and object word embeddings, θ_t and θ_o, respectively. Lastly, t_i is fed into the text encoder to generate the text features f_t.

4.3 Feature Decomposition and Fusion

To disentangle and jointly embed the representations of **triggers** and **objects** for backdoor detection, we decompose and then fuse the visual features, f_v, and the text features, f_t [42]. We first isolate how each trigger and object contributes to the text representation by averaging their respective logits. This decomposition helps the model treat triggers and objects as independent primitives, ensuring that potential backdoor triggers are not blended with the underlying objects during subsequent fusion. During training, we explicitly supervise these decomposed features to capture the semantics of each trigger and object class.

Formally, we compute the trigger and object probabilities as follows:

$$p(y = t \mid x; \theta) = \frac{\exp(f_v \cdot f_t)}{\sum_{\bar{t} \in \mathcal{T}} \exp(f_v \cdot f_t)}, \tag{5}$$

$$p(y = o \mid x; \theta) = \frac{\exp(f_v \cdot f_t)}{\sum_{\bar{o} \in \mathcal{O}} \exp(f_v \cdot f_t)}, \tag{6}$$

where $\mathcal{T}$ is the set of possible triggers, $\mathcal{O}$ is the set of possible objects, and θ denotes the learnable parameters. We then optimize cross-entropy losses for the trigger ($\mathcal{L}_{\text{tri}}$) and object ($\mathcal{L}_{\text{obj}}$) predictions:

$$\mathcal{L}_{\text{tri}} = -\frac{1}{|\mathcal{T}|}\sum_{(x,y)\in\mathcal{P}^s}\log\Big(p\big(y=(t)\mid x;\theta\big)\Big), \tag{7}$$

$$\mathcal{L}_{\text{obj}} = -\frac{1}{|\mathcal{O}|}\sum_{(x,y)\in\mathcal{P}^s}\log\Big(p\big(y=(o)\mid x;\theta\big)\Big), \tag{8}$$

where $\mathcal{P}^s$ denotes the set of **seen** triggers–object pairings.

Next, f_v and f_t are fused with a cross-attention mechanism that aligns the image and text features within a joint embedding space. Specifically, we define the query Q from f_t, and the key K and value V from f_v. The query identifies the textual aspects that need to be emphasized in the visual representation; the key–value pairs in the visual space highlight regions or features corresponding to each textual element:

$$\text{Attention}(Q,K,V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d}}\Big)V, \tag{9}$$

where d is the feature dimensionality. The result of this cross-attention is $f_{t\to v}$, a fused representation that integrates the textual context of the triggers and objects with the corresponding visual features.

4.4 Training and Inference

Our framework trains in two main stages: we first adapt the soft prompt so that the fused features $f_{t\to v}$ correctly capture the target trigger–object pairings, and then we ensure the textual representation f_t is consistent with the retrieved visual prompt. We compute the probability of a trigger–object pair $\big(t,o\big)$ by comparing the image feature f_v to the fused representation $f_{t\to v}$:

$$p_{\text{sp}}\big(y=(t,o)\mid x;\theta\big) = \frac{\exp\big(f_v\cdot f_{t\to v}\big)}{\sum_{(t',o')\in\mathcal{P}^s}\exp\big(f_v\cdot f_{t\to v}\big)}. \tag{10}$$

Minimizing the cross-entropy over these probabilities yields the soft prompt alignment loss $\mathcal{L}_{\text{sp}}$. This encourages the shifted soft prompt to correctly identify the trigger–object pairs for samples in $\mathcal{P}^s$. Next, we require that the textual representation f_t matches the retrieved pairing from the prompt repository. We define:

$$p_{\text{ret}}\big(y=(t,o)\mid x;\theta\big) = \frac{\exp\big(f_{ret}\cdot f_t\big)}{\sum_{(t',o')\in\mathcal{P}^s}\exp\big(f_{ret}\cdot f_t\big)}. \tag{11}$$

Minimizing the cross-entropy over these probabilities produces the retrieval alignment loss $\mathcal{L}_{\text{ret}}$. The total loss is a weighted sum of these components along with the prompt losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ret}} + \lambda_{\text{tri\_obj}}\Big(\mathcal{L}_{\text{tri}} + \mathcal{L}_{\text{obj}}\Big) + \lambda_{\text{sp}}\,\mathcal{L}_{\text{sp}} + \lambda_{\text{vis}}\mathcal{L}_{\text{vis}}. \tag{12}$$

During inference, the learned prompt adapter shifts the prefix tokens, the visual prompts are retrieved and averaged, and the logits are computed based on the similarity between the image and text features in the pair space. The predicted trigger–object text labels are selected by:

$$\hat{y} = \underset{(t,o)\in\mathcal{P}^{test}}{\arg\max}\; p_{\text{sp}}\big(y=(t,o)\mid x;\theta\big), \tag{13}$$

where $\mathcal{P}^{test}$ denotes the set of test trigger–object pairings, which includes seen and unseen configurations, and p_{sp} is computed following the same procedure in Eq. (10).

5 Experiments and Results

5.1 Experimental Setup

Attacks and Splits. We conduct experiments using two benchmark datasets: CIFAR-10 [49] and GTSRB [50]. CIFAR-10 contains 50,000 training images and 10,000 test images across 10 object classes, while GTSRB consists of 39,209 training images and 12,630 test images spanning 43 traffic sign classes. Recent studies [21,51] have shown that adversaries can place backdoor triggers directly on traffic signs to mislead advanced driver-assistance and autonomous-driving systems. Therefore, GTSRB provides a practical, safety-critical testbed for evaluating our proposed data-level defense system. To introduce backdoor vulnerabilities, we generate contaminated versions of all clean images using six attack patterns, while retaining the clean images themselves as an individual class. The six widely recognized backdoor attacks which are employed are: Badnets Square (Badnets-SQ) [12], Badnets Pixels (Badnets-PX) [12], Trojan Square (Trojan-SQ) [13], Trojan Watermark (Trojan-WM) [13], l_2-inv [43], and l_0-inv [43]. These attacks encompass a diverse range of backdoor characteristics, including universality, label specificity, and variations in trigger shape, size, and placement. This results in a trigger–object pairing space of 301 unique pairings for GTSRB and 70 pairings for CIFAR-10.

Implementation Details. We utilize PyTorch 1.12.1 [52] for the implementation of our model. The model is optimized using the Adam optimizer [53] and is trained over 20 epochs on the previously mentioned datasets. Both the image encoder and text encoder are based on the pretrained CLIP ViT-L/14 model, and the entire model is trained and evaluated on a single NVIDIA 2080 Ti GPU. We set $M = 20$ for both GTSRB and CIFAR-10. To assess scalability and accuracy trade-offs, all experiments are implemented with the smaller CLIP variants ViT-B/16 and ViT-B/32, repeating the same training schedule and hyperparameters.

5.2 Unseen Trigger–Object Evaluation

This experiment evaluates the performance of DBOM in both the seen (S) and unseen (U) trigger–object pairing scenarios. Specifically, the accuracy for each trigger–object pairing type is measured, assessing both the Attack (trigger) and Object classifications separately. To provide a comprehensive evaluation, we report the Harmonic Mean (HM) of the seen and unseen accuracies, which balances performance across known and novel pairings. In addition, we calculate the area under the curve (AUC), which serves as the primary metric for assessing the overall effectiveness of the model in detecting trigger-object configurations. We compare DBOM's results with CoOP [46] and CSP [40] since they represent two distinct approaches for leveraging CLIP in modeling triggers and objects as separate primitives in the embedding space. CoOP uses fixed, pre-computed natural language representations for the triggers and objects while learning only a context prompt prefix to condition CLIP. In contrast, CSP learns soft prompts by fine-tuning learnable tokens for triggers and objects, allowing for more adaptive reconfiguration and improved generalization to unseen trigger–object pairings.

Table 1 demonstrates that DBOM outperforms the baseline methods across nearly all metrics. DBOM improves AUC over 53% on GTSRB and nearly 43% on CIFAR-10. Furthermore, DBOM successfully identifies over 98% of backdoor triggers on both benchmarks while classifying nearly 95% of objects in the diverse GTSRB dataset (43 classes) and over 95% on CIFAR-10 (10 classes). Importantly, the high accuracy observed for unseen trigger-object pairings indicates that our model can detect trigger-object pairings that were **not encountered** during training. Note that DBOM not only generalizes to unseen trigger–object pairings, it also accurately identifies seen triggers: the "Seen" columns in Table 1 show over 92 and 96% accuracy on known trigger patterns.

Table 1: Comparison of backdoor trigger–object identification methods on GTSRB and CIFAR-10. Bold indicates the best results

Method	CLIP model	GTSRB						CIFAR-10					
		S	U	Att.	Obj.	HM	AUC	S	U	Att.	Obj.	HM	AUC
CoOP [46]	ViT-L/14	28.26	28.95	37.26	35.59	11.59	4.95	65.64	67.81	46.31	92.69	47.47	35.67
CSP [40]	ViT-L/14	57.34	77.86	65.27	76.85	51.07	38.03	70.28	77.81	63.34	**95.28**	62.23	50.42
DBOM (Ours)	ViT-B/32	92.65	93.70	98.31	87.10	88.05	85.03	92.09	93.76	98.19	87.38	86.76	84.43
DBOM (Ours)	ViT-B/16	93.19	95.47	**98.63**	90.32	90.21	87.86	93.40	94.90	98.31	89.51	90.22	87.37
DBOM (Ours)	ViT-L/14	**96.89**	**96.88**	98.15	**95.00**	**93.94**	**92.29**	**96.90**	**98.15**	**98.80**	95.20	**94.19**	**93.07**

Moreover, we report the results of smaller CLIP variants in Table 1 and average run-times across both datasets for each variant in Table 2. We can observe that the ViT-B/32 and ViT-B/16 models run at an average of 2.53 ms and 4.27 ms/image, compared to ViT-L/14's 10.69 ms/image, respectively. Importantly, this reduction in compute does not result in a significant drop in accuracy: the ViT-B/32–based DBOM still achieves AUC scores of 85.03% on GTSRB and 84.43% on CIFAR-10, while the ViT-B/16 variant increases those figures to 87.86% and 87.37%. These findings suggest that our approach can leverage smaller CLIP backbones for real-time deployment without sacrificing the high trigger-object identification performance afforded by the larger variant.

Table 2: Inference runtime per image on a single NVIDIA 2080 Ti GPU (batch size 64)

CLIP Variant	Inference time (ms/img)
ViT-B/32	2.53
ViT-B/16	4.27
ViT-L/14	10.69

Overall, DBOM's zero-shot generalization capability to novel trigger–object pairings is achieved by leveraging the disentangled representation learning approach, which factors triggers and objects into independent primitives. Although previous methods aim for similar generalization, our visual prompt repository, dynamic prefix adapter, feature decomposition and fusion greatly improve the ability to recombine these learned representations to accurately identify novel trigger-object pairings. Therefore, DBOM offers robust protection against evolving backdoor attack strategies by possessing the ability to identify seen configurations with high accuracy and then leveraging those seen pairings to identify unseen configurations, resulting in an adaptive method that can simultaneously evolve to adversarial strategies.

5.3 Backdoor Poison Detection Evaluation

DBOM is compared against conventional pre-training dataset cleaning approaches [20–22] by simulating a realistic scenario where the poisoning rate is set at 5%, 10%, and 15%, reflecting the poisoning ratios often encountered in web-scraped datasets. Overall accuracy (Acc.) measures the proportion of all images, both clean and poisoned, that are correctly classified. Futhermore, we report the attack recall (Rec.), indicating the percentage of poisoned images that are successfully identified. Additionally, attack precision (Prec.) measures the proportion of images flagged as attacked that are truly poisoned, and the F1 Attack score is the harmonic mean of attack precision and recall. Table 3 summarizes the performance of DBOM relative to baseline methods.

Table 3: Poison detection evaluation at 5%, 10%, and 15% poisoning levels on CIFAR-10 and GTSRB. Bold indicates the best results for each poisoning rate

Method	Poisoning rate	GTSRB				CIFAR-10			
		Acc.	Rec.	Prec.	F1	Acc.	Rec.	Prec.	F1
VisionGuard [21]	5%	88.43	57.07	23.23	33.02	85.56	48.57	16.94	25.12
	10%	85.09	62.16	35.83	45.46	88.34	65.32	44.34	52.82
	15%	90.23	63.29	68.99	66.02	90.94	70.17	69.58	69.87
Deep *k*-NN [20]	5%	**99.46**	89.13	**100.0**	94.25	98.81	76.19	**100.0**	86.49
	10%	97.11	75.35	95.65	84.40	97.59	75.90	**100.0**	86.30
	15%	94.69	64.59	**100.0**	78.48	97.45	82.95	**100.0**	90.68
HOLMES [22]	5%	95.99	35.56	96.97	52.03	**99.29**	80.00	**100.0**	88.89
	10%	96.91	69.81	**100.0**	82.22	97.53	78.43	**100.0**	87.91
	15%	93.62	57.20	99.29	72.58	97.45	83.10	**100.0**	90.77
DBOM (Proposed)	5%	98.36	**98.49**	98.83	**98.63**	97.86	**97.23**	98.86	**98.19**
	10%	**98.05**	**95.52**	98.21	**96.83**	**98.80**	**98.79**	99.05	**98.85**
	15%	**97.86**	**98.19**	98.28	**98.23**	**97.58**	**97.58**	98.06	**97.71**

Evaluation shows that DBOM consistently results in high overall accuracy while keeping the misclassification of clean samples to a minimum. For example, on GTSRB, DBOM achieves overall accuracies of around 98% with an attack recall consistently exceeding 97% and F1 scores near 98% across poisoning rates of 5%–15%. Similar trends are observed on CIFAR-10, where overall accuracies are in the range of 97%–98%, and both attack recall and F1 scores remain high. Furthermore, our experimental results reveal an important trade-off between precision and recall. While methods such as Deep *k*-NN and HOLMES achieve near perfect precision, they often suffer from lower attack recall (typically around 75%–80%), leading to significantly lower F1 scores. DBOM's modest decrease in precision is acceptable because missing a poisoned image can be far more harmful than incorrectly flagging a few additional clean images, especially when clean images make up the majority of the dataset. Lastly, unlike existing SOA methods that solely focus on identifying whether an image is backdoored or poisoned, DBOM disentangles each image's representations into primitives to identify both the trigger and the object concurrently, thereby enabling it to detect unseen configurations that were not encountered during training, a crucial improvement over existing SOA methods.

5.4 *Ablation Study*

Impact of λ_{vis}. We investigate the influence of the visual prompt loss weight, λ_{vis}, on DBOM's ability to disentangle trigger and object features. Recall that the visual prompt loss $\mathcal{L}_{\text{vis}} = \mathcal{L}_{\text{sep}} + \mathcal{L}_{\text{div}}$ enforces higher similarity for the trigger visual prompt and diversity between the trigger and object visual prompts. Note that when $\lambda_{\text{vis}} = 0.0$, the visual prompt loss is removed from the training objective and the model loses supervision to disentangle trigger and object features from the visual prompt repository, although the top two most similar prompts are still selected.

The results, shown in in Fig. 3, reveal that at $\lambda_{\text{vis}} = 0.0$, the model achieves the lowest performance across all metrics. As λ_{vis} increases, the supervision provided by the separation and diversity losses leads to improvements in both AUC and unseen accuracy, reaching a peak at $\lambda_{\text{vis}} = 0.5$. This peak indicates that a moderate emphasis on the separation losses most effectively refines the latent representations. Therefore, the model is able to generalize more robustly to unseen backdoor configurations. While selecting the top two prompts from the visual repository yields acceptable performance, incorporating the explicit separation and diversity losses significantly improves overall performance across all metrics. While results on CIFAR-10 show a more stable rise and fall of seen, unseen, and AUC values, the results on GTSRB show more variation over each tested λ_{vis} value.

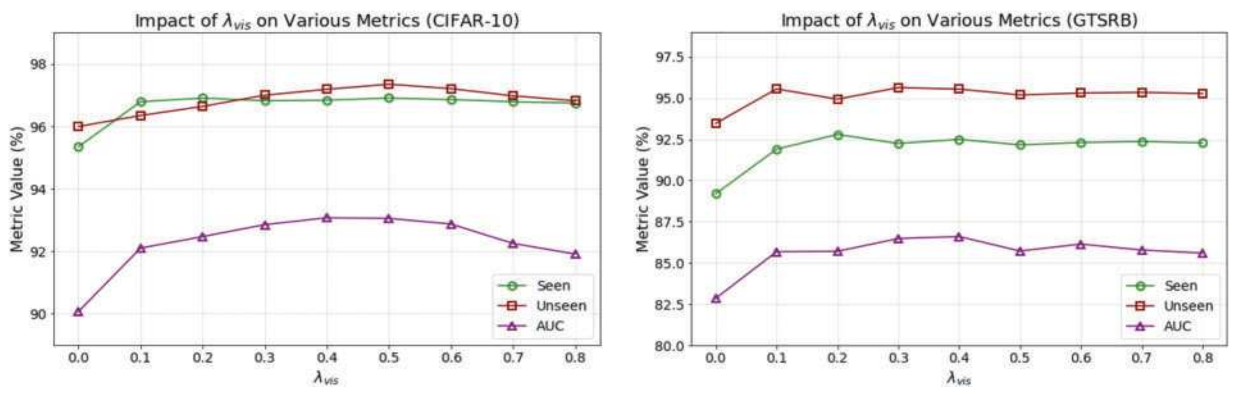

Figure 3: Impact of λ_{vis} on AUC and seen/unseen accuracy

Learnable vs. Static Prefix. In this experiment, we replace the learnable soft prompt adapter with a static fixed prompt prefix, `a photo of`, to isolate the influence of a constant prefix context on model performance. Table 4 details the performance improvement across all metrics of the learnable prefix adapter over the fixed prefix. For GTSRB, the learnable prefix leads to a 5.07% increase in object classification accuracy, AUC 3.31% and seen accuracy 2.19%. This improvement is especially significant for object classification, given that GTSRB has a diverse set of 43 classes, making the task more challenging. Similarly, on CIFAR-10, we see a notable 1.59% increase on unseen pairings, 1.38% for object classification, and 1.92% for AUC. The improvements can be attributed to dynamically adjusting the prefix tokens based on each input image's content, leading to better alignment between visual and textual representations and more precise detection. This improves the model's capability to distinguish between triggers and objects, especially when encountering unseen adversarial configurations.

Table 4: Learnable vs. Static Prefix

Method	**GTSRB**		**CIFAR-10**	
	"a photo of"	**[v1][v2][v3]**	**"a photo of"**	**[v1][v2][v3]**
Seen	90.10	92.29 (+2.19)	96.75	96.90 (+0.15)
Unseen	94.92	95.54 (+0.62)	95.75	97.34 (+1.59)
Attack	97.76	98.04 (+0.28)	97.74	98.80 (+1.06)
Object	84.09	89.16 (+5.07)	93.82	95.20 (+1.38)
HM	86.40	87.50 (+1.10)	93.01	94.19 (+1.18)
AUC	83.29	86.60 (+3.31)	91.15	93.07 (+1.92)

5.5 *Qualitative Analysis*

Fig. 4 displays randomly selected images from the test set along with the predicted trigger–object pairs and their ground-truth labels. The examples in the top row highlight *successful predictions*, illustrating how our framework can handle diverse triggers, object classes, and varying image quality. Even in blurry or distorted cases, such as the "Priority Road" sign, the model still distinguishes both the trigger and the object accurately.

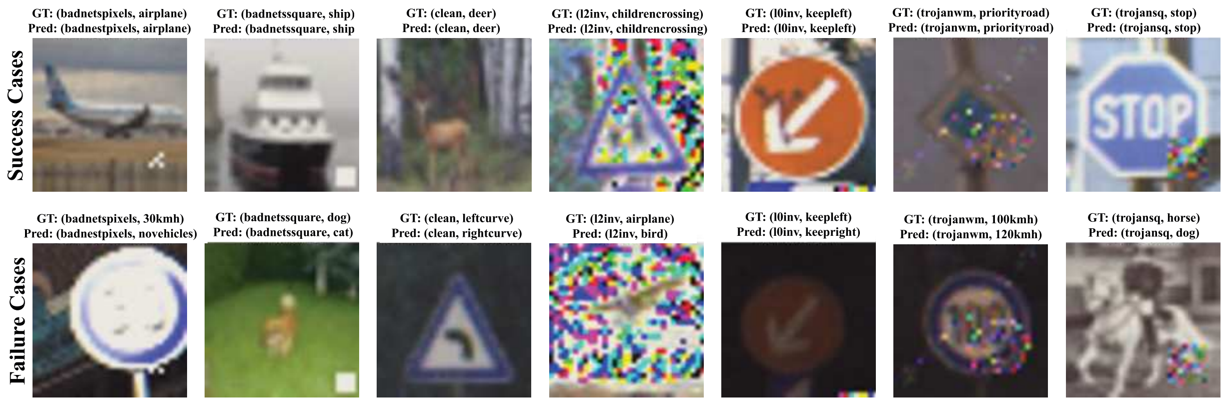

Figure 4: Ground Truth vs. Prediction of DBOM

In contrast, the bottom row depicts *failure cases* where the predicted objects differ from the ground truth (though the triggers are correctly identified). For instance, in the first error image, "30 km/h" is misclassified as "No Vehicles," likely due to the heavy blur on the sign. Likewise, in the second example, the model predicts "Dog" instead of "Cat", a plausible mistake given the animal's appearance. The fourth image is misjudged as a "Bird" rather than an "Airplane," suggesting that the system recognized a flying object but failed to capture its specific category. This can be attributed to some key features, such as text or outlines, being nearly imperceptible and making the difference between classes difficult to discern. Overall, despite a few misclassifications caused by blurred or partially obscured features, our model successfully distinguishes a wide range of triggers and objects. This highlights its strong robustness against challenging real-world conditions, even when subtle distortions could easily mislead other systems.

6 Discussion and Limitations

The empirical results demonstrate that DBOM not only achieves SOA performance in detecting both seen and unseen trigger–object pairings, but also maintains high overall accuracy and attack recall even at low poisoning rates. By proactively vetting training data, DBOM prevents backdoor contamination before downstream model training, reducing the need for costly post-training purification and preserving clean samples for model learning.

By separating the backdoor trigger from the underlying object semantics, DBOM not only flags poisoned images, but also recovers the correct object label despite the presence of a backdoor pattern. This has several key benefits. First, it preserves the majority of clean examples so that benign object information is retained rather than discarded, maintaining dataset diversity and reducing the risk of eliminating clean samples. Second, disentanglement yields finer-grained forensic insights into how specific triggers map onto different object categories, revealing systematic attacker strategies and enabling more targeted threat intelligence. Third, the modular nature of trigger and object primitives enables zero-shot detection of trigger-object pairings that were unseen during training, addressing a crucial limitation in conventional trigger-centric defenses. In practice, this means DBOM can adapt to evolving backdoor tactics across multiple object classes, lower false-positive rates by distinguishing benign from malicious features, and streamline training-time vetting helping prevent data contamination at its source rather than reactively purifying a compromised model.

Despite these strengths, it is important to discuss DBOM's limitations. Our design assumes that the defender maintains a library of T candidate trigger patterns, drawn from previously seen backdoor signatures. In our experiments, T is composed of six well-studied backdoor attacks, but the repository can be extended over time as new threats emerge by disentangling unknown triggers and adding them to the trigger repository. When novel trigger patterns are encountered in new data, we can fine-tune only the visual prompt repository and prefix adapter (rather than retraining the entire VLM backbone) on a small set of those examples, allowing DBOM to rapidly incorporate and detect new triggers with minimal overhead. Although DBOM currently focuses on triggers in T, exploring zero-shot discovery of entirely novel trigger patterns remains an important avenue for future work. Furthermore, the effectiveness of the model depends on the careful tuning of hyperparameters such as λ_{vis}, as shown in our ablation study. Moreover, DBOM is currently dependent on VLM encoders, leading to a dependency on the VLM's pre-trained weights. If the VLM fails to classify certain object classes or detect a trigger pattern, then both the visual prompt retrieval and the prefix-tuned text embedding can be skewed, leading to lower detection rates. Mitigating this risk in the future may require fine-tuning the VLM on more diverse, trigger-specific data, or swapping in more powerful multimodal backbones as they become available. However, in this manuscript, we showed base CLIP models are well adept for this task.

While our experiments so far have focused on a select set of backdoor triggers, we have not yet evaluated DBOM against adversarial perturbations generated by methods like Projected Gradient Descent (PGD) [54] or Fast Gradient Signed Method (FGSM) [55]. Such attacks work by distributing pixel-level noise within a perturbation budget: when the budget is very small, the changes are imperceptible but often yield lower attack success; when it is larger, the attack becomes more effective but also more noticeable to humans. We believe DBOM's disentangled trigger–object framework could be extended to handle perturbations with higher budgets, where the noise forms a distinct visual signature similar to the currently tested backdoor patterns and thus can cluster effectively in our visual prompt repository. In future work, we plan to explore these alternative attack types to further test DBOM's resilience. Lastly, evaluating DBOM on larger and more heterogeneous datasets and in real-world data-curation pipelines will further validate its practical utility.

7 Conclusion

In this paper, we introduced DBOM, a novel disentangled representation learning framework designed to detect both seen and unseen backdoor trigger-object pairings in training datasets. By leveraging a structured factorization of triggers and objects in the embedding space, DBOM enables robust generalization to novel backdoor configurations that evade conventional defenses. Our approach integrates a visual prompt repository and a dynamic prefix adapter to enhance the separation of adversarial triggers from underlying object representations. Experimental results demonstrate that DBOM significantly improves backdoor detection performance, outperforming SOA methods in identifying poisoned samples before they compromise downstream model training. This proactive approach not only enhances the security of DNN training pipelines but also provides deeper insights into backdoor strategies by identifying the objects associated with triggers, offering a novel method for defending against evolving backdoor threats.

Acknowledgement: We would like to acknowledge the UWF Argo Cyber Emerging Scholars (ACES) program funded by the National Science Foundation (NSF) CyberCorps® Scholarship for Service (SFS) for making this research possilble.

Funding Statement: This work is partially supported by the UWF Argo Cyber Emerging Scholars (ACES) program funded by the National Science Foundation (NSF) CyberCorps® Scholarship for Service (SFS) award under grant number 1946442. Any opinions, findings, and conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the NSF.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Kyle Stein, Andrew A. Mahyari, Guillermo Francia III; Methodology, Kyle Stein, Andrew A. Mahyari, Guillermo Francia III; Software, Kyle Stein; Validation, Kyle Stein; Formal analysis, Kyle Stein, Andrew A. Mahyari, Guillermo Francia III; Writing—original draft preparation, Kyle Stein, Andrew A. Mahyari, Guillermo Francia III; Writing—review and editing, Kyle Stein, Andrew A. Mahyari, Guillermo Francia III, Eman El-Sheikh. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, KS, upon reasonable request.

Ethics Approval: Due to the nature of backdoored attacks, all code used to generate them were drawn from methods and datasets already published by the original authors of those backdoor-attack scripts. The backdoored images and their corresponding clean counterparts used in our experiments will be made publicly available after publication to facilitate reproduction of our results. However, we will not redistribute the original attack-generation scripts. Readers may obtain those directly from the authors of the respective prior works cited in this paper.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the North American Chapter of the Association for Computational Linguistics; 2019; Minneapolis, MN, USA. p. 4171–86.
2. Chowdhary K, Chowdhary K. Natural language processing. In: Fundamentals of artificial intelligence. New Delhi, India: Springer; 2020. p. 603–49. doi:10.1007/978-81-322-3972-7_19.
3. Galassi A, Lippi M, Torroni P. Attention in natural language processing. IEEE Trans Neural Netw Learn Syst. 2020;32(10):4291–308. doi:10.1109/TNNLS.2020.3019893.
4. Li Y, Wu CY, Fan H, Mangalam K, Xiong B, Malik J, et al. Mvitv2: improved multiscale vision transformers for classification and detection, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA; 2022. p. 4794–804. doi:10.1109/cvpr52688.2022.00476.
5. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. Pattern Recognit. 2018;77(11):354–77. doi:10.1016/j.patcog.2017.10.013.
6. Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning. In: Proceeding Advanced Neural Information Process Systems, Curran Associates Inc. Red Hook, NY, USA; 2017. p. 4080–90.
7. Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access. 2018;6:14410–30. doi:10.1109/access.2018.2807385.
8. Schwinn L, Dobre D, Günnemann S, Gidel G. Adversarial attacks and defenses in large language models: old and new threats. In: Proceedings on "I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models" at NeurIPS 2023 Workshops; Cambridge, MA: PMLR; 2023. p. 103–17. [cited 2025 Jun 21]. Available from: https://proceedings.mlr.press/v239/schwinn23a.html.

9. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. Cambridge, MA: PMLR; 2021. p. 5028748–63.
10. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM. 2020;63(11):139–44. doi:10.1145/3422622.
11. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022; New Orleans, LA, USA. p. 10684–95. doi:10.1109/cvpr52688.2022.01042.
12. Gu T, Liu K, Dolan-Gavitt B, Garg S. BadNets: evaluating backdooring attacks on deep neural networks. IEEE Access. 2019;7:47230–44. doi:10.1109/access.2019.2909068.
13. Liu Y, Ma S, Aafer Y, Lee WC, Zhai J, Wang W, et al. Trojaning attack on neural networks. In: 25th Annual Network And Distributed System Security Symposium (NDSS 2018). San Diego, CA, USA. Reston, VA: Internet Soc; 2018. doi:10.14722/ndss.2018.23291.
14. Li YL, Xu Y, Mao X, Lu C. Symmetry and group in attribute-object compositions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Seattle, WA, USA. p. 11316–25. doi:10.1109/cvpr42600.2020.01133.
15. Nguyen A, Tran A. Wanet–imperceptible warping-based backdoor attack. arXiv:2102.10369. 2021.
16. Niu Y, He S, Wei Q, Wu Z, Liu F, Feng L. Bdetclip: multimodal prompting contrastive test-time backdoor detection. arXiv:2405.15269. 2024.
17. Guo M, Yang Y, Xu R, Liu Z, Lin D. When nas meets robustness: in search of robust architectures against adversarial attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020; Seattle, WA, USA. p. 631–40. doi:10.1109/cvpr42600.2020.00071.
18. Yue Z, Lin B, Zhang Y, Liang C. Effective, efficient and robust neural architecture search. In: 2022 International Joint Conference on Neural Networks (IJCNN). Padua, Italy: IEEE; 2022. p. 1–8.
19. Zhu M, Wei S, Shen L, Fan Y, Wu B. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023; Paris, France. p. 4466–77. doi:10.1109/iccv51070.2023.00412.
20. Peri N, Gupta N, Huang WR, Fowl L, Zhu C, Feizi S, et al. Deep k-nn defense against clean-label data poisoning attacks. In: Computer Vision–ECCV 2020 Workshops; 2020 Aug 23–28; Glasgow, UK: Springer; 2020. p. 55–70. doi:10.1007/978-3-030-66415-2_4.
21. Kantaros Y, Carpenter T, Sridhar K, Yang Y, Lee I, Weimer J. Real-time detectors for digital and physical adversarial inputs to perception systems. In: Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems. Nashville, TN, USA; 2021. p. 67–76. doi:10.1145/3450267.3450535.
22. Wen J. HOLMES: to detect adversarial examples with multiple detectors. arXiv:2405.19956. 2024.
23. Stein K, Mahyari AA, Francia G, El-Sheikh E. Proactive adversarial defense: harnessing prompt tuning in vision-language models to detect unseen backdoored images. arXiv:2412.08755. 2024.
24. Karim N, Arafat AA, Khalid U, Guo Z, Rahnavard N. Augmented neural fine-tuning for efficient backdoor purification. In: European Conference on Computer Vision. Cham, Switzerland: Springer Nature Switzerland; 2024. p. 401–18. doi:10.1007/978-3-031-72989-8_23.
25. Xu Y, Gu Y, Sakurai K. PAD-FT: a lightweight defense for backdoor attacks via data purification and fine-tuning. arXiv:2409.12072. 2024.
26. Chen B, Carvalho W, Baracaldo N, Ludwig H, Edwards B, Lee T, et al. Detecting backdoor attacks on deep neural networks by activation clustering. arXiv:1811.03728. 2018.
27. Tran B, Li J, Madry A. Spectral signatures in backdoor attacks. Adv Neural Inf Process Syst. 2018;31. doi:10.48550/arXiv.1811.00636.
28. Hayase J, Kong W, Somani R, Oh S. SPECTRE: defending against backdoor attacks using robust statistics. In: Proceedings of the 38th International Conference on Machine Learning. Cambridge, MA: Proceedings of Machine Learning Research (PMLR); 2021. p. 4129–39. [cited 2025 Jun 21]. Available from: https://proceedings.mlr.press/v139/hayase21a.html.

29. Tang D, Wang X, Tang H, Zhang K. Demon in the variant: statistical analysis of DNNs for robust backdoor contamination detection. In: 30th USENIX Security Symposium (USENIX Security 21). Berkeley, CA: USENIX Association; 2021. p. 1541–58. [cited 2025 Jun 21]. Available from: https://www.usenix.org/conference/usenixsecurity21/presentation/tang-di.
30. Ma W, Wang D, Sun R, Xue M, Wen S, Xiang Y. The "Beatrix" resurrections: robust backdoor detection via gram matrices. arXiv:2209.11715. 2022.
31. Li Y, He J, Huang H, Sun J, Ma X. Shortcuts everywhere and nowhere: exploring multi-trigger backdoor attacks. arXiv:2401.15295. 2024.
32. Lake BM. Towards more human-like concept learning in machines: compositionality, causality, and learning-to-learn. massachusetts institute of technology; 2014. [cited 2025 Jul 10]. Available from: http://hdl.handle.net/1721.1/95856.
33. Tong B, Wang C, Klinkigt M, Kobayashi Y, Nonaka Y. Hierarchical disentanglement of discriminative latent features for zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA; 2019. p. 11467–76.
34. Chen Z, Luo Y, Qiu R, Wang S, Huang Z, Li J, et al. Semantics disentangling for generalized zero-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021; Montreal, QC, Canada. p. 8712–20. doi:10.1109/iccv48922.2021.00859.
35. Li X, Xu Z, Wei K, Deng C. Generalized zero-shot learning via disentangled representation. Proc AAAI Conf Artif Intell. 2021;35(3):1966–74. doi:10.1609/aaai.v35i3.16292.
36. Hao S, Han K, Wong KYK. Learning attention as disentangler for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 18–24; Vancouver, BC, Canada. p. 15315–24. doi:10.1109/cvpr52729.2023.01470.
37. Stein K, Mahyari A, Francia G, El-Sheikh E. Visual adaptive prompting for compositional zero-shot learning. arXiv:2502.20292. 2025.
38. Li X, Yang X, Wei K, Deng C, Yang M. Siamese contrastive embedding network for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022; Orleans, LA, USA. p. 9326–35. doi:10.1109/cvpr52688.2022.00911.
39. Jing C, Li Y, Chen H, Shen C. Retrieval-augmented primitive representations for compositional zero-shot learning. Proc AAAI Conf Artif Intell. 2024;38(3):2652–60. doi:10.1609/aaai.v38i3.28043.
40. Nayak NV, Yu P, Bach SH. Learning to compose soft prompts for compositional zero-shot learning. arXiv:2204.03574. 2022.
41. Xu G, Chai J, Kordjamshidi P. GIPCOL: graph-injected soft prompting for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2024; Waikoloa, HI, USA. p. 5774–83. doi:10.1109/wacv57701.2024.00567.
42. Lu X, Guo S, Liu Z, Guo J. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023; Vancouver, BC, Canada. p. 23560–9. doi:10.1109/cvpr52729.2023.02256.
43. Li S, Xue M, Zhao BZH, Zhu H, Zhang X. Invisible backdoor attacks on deep neural networks via steganography and regularization. IEEE Trans Dependable Secure Comput. 2020;18(5):2088–105. doi:10.1109/tdsc.2020.3021407.
44. Wu B, Wei S, Zhu M, Zheng M, Zhu Z, Zhang M, et al. Defenses in adversarial machine learning: a survey. arXiv:2312.08890. 2023.
45. Avd O, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv:1807.03748. 2018.
46. Zhou K, Yang J, Loy CC, Liu Z. Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 17–24; New Orleans, LA, USA. p. 16816–25. doi:10.1109/cvpr52688.2022.01631.
47. Zhou K, Yang J, Loy CC, Liu Z. Learning to prompt for vision-language models. Int J Comput Vis. 2022;130(9):2337–48. doi:10.1007/s11263-022-01653-1.

48. Khattak MU, Rasheed H, Maaz M, Khan S, Khan FS. Maple: multi-modal prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 19113–22. doi:10.1109/cvpr52729.2023.01832.
49. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. 2009. [cited 2025 Jul 10]. Available from: https://www.cs.utoronto.ca/.
50. Stallkamp J, Schlipsing M, Salmen J, Igel C. Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw. 2012;32(1):323–32. doi:10.1016/j.neunet.2012.02.016.
51. Yao Y, Li H, Zheng H, Zhao BY. Latent backdoor attacks on deep neural networks. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security; 2019 Nov 06; London, UK. p. 2041–55.
52. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst. 2019;32. doi:10.5555/3454287.3455008.
53. Kingma DP. Adam: a method for stochastic optimization. arXiv:1412.6980. 2014.
54. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations, ICLR 2018; 2018 Apr 30–May 3; Vancouver, BC, Canada.
55. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Bengio Y, LeCun J, editors. In: 3rd International Conference on Learning Representations, ICLR 2015; 2015 May 7–9; San Diego, CA, USA.